Concept 2.6
Manuel de l'utilisateur

Volume 1

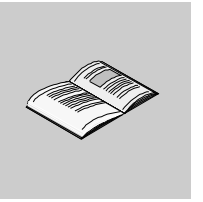
11/2007

33002205.10

Merlin Gerin
Square D
Telemecanique

Schneider Electric

Table des matières



Consignes de sécurité XI

A propos de ce manuel XIII

Chapitre 1 Description générale de Concept 1

Introduction 1

1.1 Description générale de Concept 3

Introduction 3

Introduction 4

Configuration du matériel API 6

Contenu de paquet du matériel API en Concept S, M et XL 7

1.2 Programmation 9

Introduction 9

Remarques générales 10

Bibliothèques 12

Editeurs 15

Fonctions en ligne 20

Communication 21

Application protégée 22

Programmes utilitaires 24

Chapitre 2 Nouveautés de Concept 2.6 par rapport à Concept 2.5 29

Introduction 29

Nouvelles fonctionnalités de Concept 2.6 par rapport à Concept 2.5 30

Nouvelles fonctionnalités de Concept 2.6 SR2
par rapport à Concept 2.6 SR1 36

Nouvelles fonctionnalités de Concept 2.6 SR3
par rapport à Concept 2.6 SR2 39

Chapitre 3 Structure des projets 41

Introduction 41

Structure et traitement des projets 42

Programme 49

Sections 54

Données de configuration 60

Chapitre 4 Créer un projet 61
Introduction 61
Vue d'ensemble 62
Etape 1 : démarrage de Concept 63
Etape 2 : Configuration automate 64
Etape 2.1 : Configuration requise 65
Etape 2.2 : Configuration optionnelle 68
Etape 3 : création du programme utilisateur 72
Etape 4 : Enregistrer 75
Etape 5 : Effectuez l' évaluation mémoire 76
Etape 6 : Chargement et test 77
Etape 7 : Optimisation et séparation 82
Etape 8 : Documentation 84

Chapitre 5 Configuration API 87
Introduction 87
5.1 Remarques générales sur la configuration matérielle 89
Introduction 89
Remarques générales 90
Procédez ainsi lors de la configuration 91
5.2 Configuration en mode local et connecté 93
Introduction 93
Remarques générales 94
Fonctions disponibles en mode EN LIGNE et HORS LIGNE 95
5.3 Configuration inconditionnelle 97
Introduction 97
Condition préalable 98
Sélection automate 99
Sélection d'UC pour le type d'automate 100
Partition de mémoire API 105
Instructions chargeables 106
Ordonnanceur de segments 109
Affectation des E/S 111
5.4 Configuration optionnelle 115
Introduction 115
Configuration des informations ASCII 116
Rendre disponibles d'autres fonctions dans le configurateur 117
Echange de données entre abonnés au réseau Modbus Plus 118
Nombre de mots vraiment utilisés lors de la
réception de données (Peer Cop) 119
Protéger les données contre l'accès dans la mémoire de signal 121
Paramétrage des interfaces 122
Options diverses 124

5.5 Backplane Expander Config 127
Vue d'ensemble. 127
Généralités relatives à l'extenseur d'arrière-plan 128
Modifier la répartition E/S 129
Traitement des erreurs 130
5.6 Configuration de différents systèmes de réseau 131
Introduction 131
Configurer système INTERBUS 132
Configurer le système DP Profibus 133
Configurer Ethernet 135
Extension RTU 137
Scrutateur E/S / Ethernet 138
Comment utiliser le scrutateur E/S / Ethernet 142
5.7 Paramètres de sécurité Quantum dans le configurateur. 145
Paramètres de sécurité Quantum 145

Chapitre 6 Structure schématique de la mémoire de l'API et optmisation de la mémoire. 149

Introduction 149
6.1 Structure schématique de la mémoire de l'API 151
Structure par principe de la mémoire de l'API. 151
6.2 Remarques générales sur l'optimisation de la mémoire 153
Introduction 153
Possibilités d'optimisation de la mémoire 154
Ne dépendant pas de l'automate 155
6.3 Optimiser la mémoire pour l'UC X13 0X et 424 02 Quantum 159
Introduction 159
Généralités sur l'optimisation de la mémoire pour l'UC X13 0X et 424 02 Quantum 160
Sélectionner le fichier EXEC optimal. 162
Utiliser la mémoire étendue (mémoire d'état pour les références 6x) 166
Synchronisation de la zone CEI et de la zone LL984 168
Synchronisation des zones pour les données globales et la mémoire du programme CEI 170
6.4 Optimisation de la mémoire pour l'UC 434 12(A) et 534 14(A/B) Quantum 173
Introduction 173
Généralités sur l'optimisation de la mémoire pour l'UC 434 12(A) et 534 14(A/B) Quantum 174
Synchronisation de la zone CEI et de la zone LL984 176
Synchronisation des zones pour les données globales et la mémoire du programme CEI (UC 434 12(A) / 534 14(A/B)) 181

6.5 Optimisation de mémoire pour les UC Compact 185
Introduction 185
Généralités sur l'optimisation de la mémoire pour les UC Compact 186
Synchronisation de la zone CEI et de la zone LL984 188
Synchronisation des zones pour les données globales et la mémoire du programme CEI (Compact) 193
6.6 Optimisation de mémoire pour les UC Momentum 197
Introduction 197
Généralités sur l'optimisation de la mémoire pour les UC Momentum 198
Sélectionner le fichier EXEC optimal 200
Synchronisation des zones pour les données globales et la mémoire du programme CEI (Momentum) 201
6.7 Optimisation de mémoire pour les UC Atrium 205
Introduction 205
Généralités sur l'optimisation de la mémoire pour les UC Atrium 206
Utilisation de CEI 207
Synchronisation des zones pour les données globales et la mémoire du programme CEI (Atrium) 208

Chapitre 7 Langage des blocs fonctions FDB 211
Introduction 211
7.1 Remarques générales sur le langage du bloc fonction FDB 213
Remarques générales sur le langage du bloc fonction FDB 213
7.2 Objets du langage de bloc fonction FBD 214
Introduction 214
Fonctions et blocs fonctions (FFB) 215
Liaison 219
Paramètres réels 220
Objet texte 222
7.3 Tâches exécutées en langage de bloc fonction FBD 223
Introduction 223
Placement de fonctions et de blocs fonctions 224
Ordre d'exécution des FFB 226
Configuration de boucles 228
7.4 Génération de codes dans le cadre du langage de bloc fonction FBD 229
Options de génération de code 229
7.5 Fonctions en ligne du langage de bloc fonction FBD 231
Fonctions en ligne 231
7.6 Elaboration d'un programme avec le langage de bloc fonction FBD 233
Etablissement d'un programme dans le langage de bloc fonction FBD 233

Chapitre 8 Plan de contact LD 237
Introduction 237
8.1 Remarques générales sur le langage à contacts LD 239
Remarques générales sur le langage du schéma à contacts LD 239

8.2 Objets dans le langage à contacts LD. 241
Introduction 241
Contacts 242
Bobines 244
Fonctions et blocs fonctions (FFB) 247
Liaison 253
Paramètres réels 254
Objet texte 256
8.3 Tâches exécutées en langage à contacts LD 257
Introduction 257
Placement de bobines, de contacts, de fonctions et de blocs fonctions 258
Ordre d'exécution 260
Configuration de boucles. 262
8.4 Génération de codes dans le cadre du langage à contacts LD. 263
Options de génération de code 263
8.5 Fonctions en ligne dans le cadre du langage à contacts LD. 265
Fonctions en-ligne 265
8.6 Elaboration d'un programme avec le langage à contacts LD 268
Etablissement d'un programme dans le plan de contact LD 268

Index i

Les chapitres grisés ne font pas partie de ce volume.

Chapitre 9 Langage de commande séquentielle SFC 273

Chapitre 10 Liste d'instructions IL 325

Chapitre 11 Texte structuré ST 399

Chapitre 12 Ladder Logic 984 463

Chapitre 13 DFB (blocs fonctions dérivés) 491

Chapitre 14 Macros 535

Chapitre 15 Editeur de variables 559

Chapitre 16 Navigateur de projet 573

Chapitre 17 Types de données dérivés 581

Chapitre 18 Editeur des données de référence 611

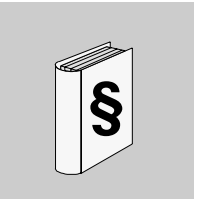
Chapitre 19 Editeur de messages ASCII 627

Chapitre 20 Fonctions en ligne 649
Chapitre 21 Import/Export 713
Chapitre 22 Documentation et archivage 757
Chapitre 23 Simulation d'un API 773
Chapitre 24 Concept-Security 787

Annexes 805
Annexe A Tableaux des performances dépendant de l'API 807
Annexe B Interface Windows 833
Annexe C Barre d'outils et touches rapides 855
Annexe D Conformité CEI 887
Annexe E Exemples de configuration 917
Annexe F Convertir Projets/DFB/Macros 1023
Annexe G Concept ModConnect 1027
Annexe H Conversion des programmes Modsoft 1037
Annexe I Références Modsoft et 984 1043
Annexe J Préréglages en cas de l'utilisation de Modbus Plus pour la première mise en service 1047
Annexe K Préréglages en cas de l'utilisation de Modbus pour la première mise en service 1061
Annexe L Première mise en service en cas d'utilisation de Modbus avec l'EXECLoader 1067
Annexe M Première mise en service dans le cas de l'utilisation de Modbus avec le DOS-Loader 1089
Annexe N Première mise en service en cas d'utilisation de Modbus Plus avec l'EXECLoader 1103

Annexe O Première mise en service dans le cas de l'utilisation de Modbus Plus avec le DOS-Loader 1125
Annexe P Fichiers EXEC 1143
Annexe Q Fichiers INI 1147
Annexe R Traitement d'interruption 1167
Annexe S Connexion automatique à l'API 1197

Consignes de sécurité



Informations importantes

AVIS

Veuillez lire soigneusement ces consignes et examiner l'appareil afin de vous familiariser avec lui avant son installation, son fonctionnement ou son entretien. Les messages particuliers qui suivent peuvent apparaître dans la documentation ou sur l'appareil. Ils vous avertissent de dangers potentiels ou attirent votre attention sur des informations susceptibles de clarifier ou de simplifier une procédure.

L'apposition de ce symbole à un panneau de sécurité Danger ou Avertissement signale un risque électrique pouvant entraîner des lésions corporelles en cas de non-respect des consignes.

Ceci est le symbole d'une alerte de sécurité. Il vous avertit d'un risque de blessures corporelles. Respectez scrupuleusement les consignes de sécurité associées à ce symbole pour éviter de vous blesser ou de mettre votre vie en danger.

⚠ DANGER

DANGER indique une situation immédiatement dangereuse qui, si elle n'est pas évitée, **entraînera** la mort ou des blessures graves.

⚠ AVERTISSEMENT

AVERTISSEMENT indique une situation présentant des risques susceptibles de **provoquer** la mort, des blessures graves ou des dommages matériels.

⚠ ATTENTION

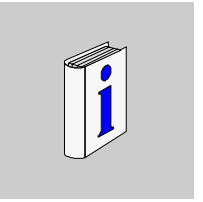
ATTENTION indique une situation potentiellement dangereuse et susceptible d'**entraîner** des lésions corporelles ou des dommages matériels.

REMARQUE IMPORTANTE

Les équipements électriques doivent être installés, exploités et entretenus par un personnel d'entretien qualifié. Schneider Electric n'assume aucune responsabilité des conséquences éventuelles découlant de l'utilisation de cette documentation.

© 2007 Schneider Electric. Tous droits réservés.

A propos de ce manuel



Présentation

Objectif du document

Ce manuel de l'utilisateur vous aidera à configurer un programme utilisateur avec Concept. Il vous offre des informations détaillées sur les différents langages de programmation ainsi que sur la configuration matérielle.

Champ d'application

Cette documentation s'applique à Concept 2.6 sous Microsoft Windows 98, Microsoft Windows 2000, Microsoft Windows XP et Microsoft Windows NT 4.x.

Note : Vous trouverez des indications supplémentaires récentes dans le fichier README de Concept.

Document à consulter

Titre	Référence
Introduction à l'installation de Concept	840 USE 502 01
Bibliothèque de modules Concept CEI	840 USE 504 01
Manuel de l'utilisateur Concept-EFB	840 USE 505 00
Bibliothèque de modules Concept LL984	840 USE 506 01

Commentaires utilisateur

Envoyez vos commentaires à l'adresse e-mail techpub@schneider-electric.com

Description générale de Concept

1

Introduction

Aperçu

Ce chapitre contient une description générale de Concept. Son objectif est de vous fournir un premier aperçu général sur Concept et ses programmes utilitaires.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
1.1	Description générale de Concept	3
1.2	Programmation	9

1.1 Description générale de Concept

Introduction

Aperçu

Ce chapitre contient une description des caractéristiques de puissance de Concept et vous fournit un aperçu général sur le matériel programmable à l'aide de Concept.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Introduction	4
Configuration du matériel API	6
Contenu de paquet du matériel API en Concept S, M et XL	7

Introduction

Système d'exploitation

L'exécution des tâches de ce genre requiert à l'heure actuelle une interface utilisateur graphique. Le résultat est la création de Concept en tant qu'application pour MS-Windows. Concept peut être utilisé sous Windows 98, Windows 2000, Windows XP et Windows NT. L'avantage de ces systèmes d'exploitation réside dans le fait que ce sont les systèmes les plus répandus au monde, et que la tecnique de fenêtres et le maniement d'une souris font partie des connaissances élémentaires d'un utilisateur d'ordinateur. En outre, MS-Windows permet d'utiliser tous les écrans, toutes les cartes graphiques et imprimantes d'usage courant. En tant qu'utilisateur, vous n'êtes donc pas tenus de choisir une configuration de matériel précise.

Norme internationale CEI 1131-3

Concept fournit un environnement autonome pour une projection matérielle effective conformément à la norme internationale CEI 1131-3.

Autonomie d'API lors de la programmation

Le slogan attitré lors du développement de Concept fut le principe du même Look and Feel, c'est-à-dire que le travail au niveau de toutes les étapes de la projection matériel et de tous les éditeurs se fait de manière intuitive de manière à ce qu'un seul coup d'il à l'écran suffise pour savoir ce qu'il faut (ne faut pas) faire. La plupart des étapes de projection, la création du programme en particulier, ont des structures indépendantes de l'API à programmer.

Représentation graphique

Le programme entier est subdivisé en sections conformément à la structure logique.

Grâce à l'outil de projection Concept, vous pouvez facilement Sélectionner, Placer et Déplacer des objets (par exemple les blocs fonctions, étapes, transitions) sous forme graphique. Dans l'éditeur SFC (Sequential Function Chart / langage de la commandeséquentielle), les contrôles de vraisemblance se font déjà lors du placement des objets, puisque la plupart des connexions entre objets sont automatiquement établies lors du placement. Dans les éditeurs FBD (Function Block Diagram) et LD (Ladder Diagram), les contrôles de vraisemblance ont lieu lors de la connexion des blocs. Les connexions qui ne sont pas permises, telles des connexions entre différents types de données, sont refusées déjà lors du projet. Dans l'éditeur LL984 (Schéma à contacts 984), un contrôle de vraisemblance a également lieu lors du placement. Dans les éditeurs IL (Instruction List) et ST (Structured Text), les instructions qui ne sont pas permises sont marquées par un encadrement en couleur. Après le premier déroulement normal du programme, le programme crée peut être optimisé sur le plan graphique en améliorant la représentation grâce au déplacement de connexions, de blocs fonctions ou de textes.

Imprimer

Sur demande, les chapitres sont représentés avec indication du format d'impression, de telle manière qu'on puisse contrôler la représentation sur chaque feuille de la documentation. Les signaux reçoivent une désignation complète avec nom de symbole et commentaire. Sur les lacunes, des indications précises sont données qui permettent de tracer le signal. Dans l'éditeur FBD, l'ordre d'exécution des blocs individuels dans une section peut être affiché et documenté.

Fonctions Import / Export

Les fonctions Import/Export permettent de regrouper à volonté des sections de différents projets en un nouveau projet.

Il est également possible de convertir des sections d'un langage de programmation CEI en des sections d'un autre langage de programmation CEI.

Les variables peuvent être exportées et importées dans le texte dans le format texte avec séparateur ou FactoryLink.

Système d'exécution

Le Système d'exécution d'API permet de réagir rapidement aux modifications du processus (court temps de cycle), Simulation des sources de signaux (voir *Simulation d'un API, p. 773*), Affichage en ligne (voir *Fonctions en ligne, p. 649*) des états, modifications en ligne de paramètres et programmes.

Structure logiciel ouverte

Concept possède une structure logiciel ouverte qui permet d'exploiter des systèmes externes (par exemple de visualisation) via interfaces standards.

Aide en ligne

Un soin particulier fut porté au développement de la fonction d'aide. La fonction Aide en ligne (voir *Structure de l'aide en ligne, p. 852*) propose une aide contextuelle dans chaque situation de la projection qu'on peut obtenir en cliquant une fois avec la souris ou en appuyant sur la touche F1. Cette aide contextuelle est disponible aussi bien pour les commandes de menu et les dialogues que pour les fonctions, les blocs fonctions ainsi que les matériels dans quelques gammes d'automate.

Configuration du matériel API

Description

Concept est l'outil de projection standard pour les produits Quantum, Compact, Momentum et Atrium.

La configuration des matériels (par exemple unité centrale, mémoire programme, unités d'entrée/sortie etc.) peut être réalisée avant, pendant ou après la création du programme.

Cette tâche de projection peut avoir lieu aussi bien avec l'API en mode connecté (Connecté) qu'en mode Hors ligne (terminal seul). Concept supporte la projection en proposant uniquement les combinaisons autorisées. ce qui permet d'empêcher efficacement une projection erronée. En mode connecté, il est aussitôt procédé au contrôle de vraisemblance du matériel projeté et les entrées erronées sont refusées.

Après la connexion du terminal (PC) avec l'API, il est procédé au contrôle de vraisemblance des valeurs projetées (par exemple de l'éditeur de variables) avec les ressources matériel réelles, et un message d'erreur est affiché, le cas échéant.

Contenu de paquet du matériel API en Concept S, M et XL

Description

Contenu de paquet du matériel API en Concept S, M et XL:

Version de Concept	**contenez le matériel**
Concept Vx.x **S**	Momentum
Concept Vx.x **M**	Compact, Momentum
Concept Vx.x **XL**	Atrium, Compact, Momentum, Quantum

1.2 Programmation

Introduction

Aperçu

Ce chapitre vous donne un aperçu général sur les éditeurs disponibles dans Concept.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Remarques générales	10
Bibliothèques	12
Editeurs	15
Fonctions en ligne	20
Communication	21
Application protégée	22
Programmes utilitaires	24

Remarques générales

Introduction

Pour accomplir les tâches dans le domaine de la technique de commande et de réglage, Concept met à votre disposition les langages de programmation conformes à CEI 1131-3 suivantes :

- Langage blocs fonctions FBD (Function Block Diagram) (voir *Editeur FDB, p. 16*),
- Plan de contact LD (Ladder Diagram) (voir *Editeur LD, p. 16*),
- Langage de commande séquentielle SFC (Sequential Function Chart) (voir *Editeur SFC, p. 17*),
- Liste d'instructions IL (Instruction List) (voir *Editeur IL, p. 17*) et
- Texte structuré ST (Structured Text) (voir *Editeur ST, p. 18*).

Le langage orienté Modsoft est également disponible

- Plan de contact LL984 (Ladder Logic) (voir *Editeur LL984, p. 18*).

Les éléments de base des langages de programmation CEI (FDB, LD, SFC, ST et IL) sont les Fonctions et les Blocs fonctions dont l'ensemble constitue des unités logiques. Concept contient diverses Bibliothèques des blocs (voir *Bibliothèques, p. 12*) qui renferment des fonctions/blocs fonctions (EFB) élémentaires déjà prédéfinis. Celles-ci sont réparties dans différents groupes suivant leur domaine d'application, ce qui facilite la recherche des EFB individuels.

Pour LL984, le langage de programmation orientée vers Modsoft, une Bibliothèque des blocs (voir *Bibliothèques, p. 12*) avec des Instructions est disponible.

Sections

Le programme de commande est constitué par des sections qui correspondent à sa structure logique. A l'intérieur d'une section, on travaille avec un seul langage de programmation.

De la jonction de ces sections naît un ensemble qui constitue le programme de commande suivant lequel l'automate commande le déroulement du processus. A l'intérieur d'un programme, les sections CEI (FDB, LD, SFC, IL, ST) peuvent être mélangées à volonté. Les sections LL984 sont toujours traitées sous forme de blocs avant les sections CEI.

Type de données

Un sous-ensemble de Types de données de la norme internationale CEI 1131-3 est à votre disposition.

Par l'Editeur du type de données (voir *Editeur de type de données (Editeur DDT), p. 19*), vous pouvez dériver vos propres types de données à partir des types de données CEI.

Utilisation de variables

Des variables ne sont généralement pas utilisées pour la liaison d'objets au sein d'une section dans le cas des langages de programmation graphiques FDB, LD, SFC et LL984, puisque ces liaisons sont réalisées à travers des connexions graphiques. (Une liaison supplémentaire avec variables est uniquement nécessaire pour les sections très complexes). Les connexions graphiques sont gérées par le système et n'entraînent par conséquent aucun effort de projection pour vous. Toutes les variables restantes, telles que les variables pour la transmission des données entre différents sections, sont projetées avec l'Editeur de variables (voir *Editeur de variables, p. 19*).

Bibliothèques

Introduction

Concept met à votre disposition différentes bibliothèques de blocs avec des Fonctions et des Blocs fonctions déjà prédéfinies pour la création du programme.

On distingue entre deux types de bibliothèques de blocs :

- Bibliothèques CEI
  Bibliothèques des blocs pour les sections dans les langages de programmation CEI (FDB, LD, SFC, IL et ST)
- Bibliothèque LL984
  Bibliothèque des blocs pour les sections dans le langage de programmation LL984 orientée vers Modsoft

Bibliothèques CEI

Les bibliothèques CEI suivantes sont disponibles pour vos applications :

- **AKFEFB**
  Cette bibliothèque contient les EFB de AKF/ALD qui ne sont pas couverts par la bibliothèque CEI.
- **ANA_IO**
  Cette bibliothèque permet de traiter des valeurs analogiques.
- **COMM**
  Cette bibliothèque permet de partager des informations entre un API et un autre abonné Modbus, Modbus Plus ou Ethernet.
- **CONT_CTL**
  Cette bibliothèque permet la projection des circuits de régulation technologiques. Elle contient les EFB régulateurs, dérivateurs, d'intégration et à courbes polygonales.
- **DIAGNOSTIC**
  Cette bibliothèque permet d'établir un diagnostic du programme de commande après des défaillances. Elle contient des EFB de contrôle de diagnostics d'action, de réaction, de blocage, des conditions de processus, des diagnostics dynamiques et des groupes de signaux.
- **EXPERTS**
  Cette bibliothèque contient les EFB nécessaires pour l'utilisation des modules d'experts.
- **EXTENDED**
  Cette bibliothèque contient des accessoires utiles qui complètent les différentes bibliothèques. Elle contient les EFB nécessaires au calcul de valeur moyenne, à la sélection de la valeur maximale, à la fonction NON, au déclenchement, à la conversion, à la formation d'un cheminement polygonal avec interpolation d'ordre 1, à l'identification de la décroissance et à la définition d'une bande morte pour les variables commandées.
- **FUZZY**
  Cette bibliothèque contient des EFB pour la logique analogique (Fuzzy).
- **CEI**
  Cette bibliothèque contient les EFB définis dans la norme internationale CEI 1131-3. Elle contient par exemple les EFB pour les fondements des mathématiques, le compteur, la temporisation etc.
- **LIB984**
  Cette bibliothèque contient les EFB conforme à CEI 1131 de la bibliothèque LL984, par exemple les EFB pour le transfert de registres.
- **SYSTEM**
  Cette bibliothèque contient les EFB pour l'utilisation des fonctions système. Elle dispose des EFB pour la reconnaissance du temps de cycle, l'utilisation de différentes horloges système, la commande de sections SFC et l'affichage de l'état de système.

Bibliothèque LL984

La bibliothèque LL984 contient les instructions (blocs) pour l'éditeur LL984. Elle contient des instructions pour les fondements des mathématiques, le compteur, la temporisation, des instructions pour l'affichage de l'état de système, des instructions de réglage, de différenciation et d'intégration ainsi que des instructions pour le partage des informations entre un API et un autre abonné Modbus ou Modbus Plus.

Editeurs

Introduction

En créant une section, vous déterminez le langage de programmation que vous désirez utiliser.

Pour la création de sections dans les différentes langages de programmation, des éditeurs spéciaux sont disponibles :

- Editeur FDB (voir *Editeur FDB, p. 16*)
- Editeur LD (voir *Editeur LD, p. 16*)
- Editeur SFC (langage de commande séquentielle) (voir *Editeur SFC, p. 17*)
- Editeur IL (voir *Editeur IL, p. 17*)
- Editeur ST (voir *Editeur ST, p. 18*)
- Editeur LL984 (plan de contact similaire à Modsoft) (voir *Editeur LL984, p. 18*)

Pour la déclaration de variables, la création de types de données et la visualisation de variables, les éditeurs suivants sont disponibles :

- L'éditeur de variables (pour déclarer les variables), (voir *Editeur de variables, p. 19*)
- L'éditeur des données de référence (pour visualiser et changer les valeurs en Connecté) (voir *Editeur des données de référence, p. 19*) et
- L'éditeur de type de données (pour créer des types de données définis par l'utilisateur) (voir *Editeur de type de données (Editeur DDT), p. 19*).

Pour la création des fonctions et blocs fonctions définis par l'utilisateur, les éditeurs suivants sont à votre disposition :

- Concept-DFB (pour créer des blocs fonctions et macros dérivés) (voir *Concept-DFB, p. 25*)
- Concept-EFB (pour créer des fonctions élémentaires et blocs fonctions définis par l'utilisateur) (voir *Concept-EFB, p. 26*)

Editeur FDB

L'Editeur FDB (voir *Langage des blocs fonctions FDB, p. 211*) permet la programmation graphique du plan de fonctionnement conformément à la norme CEI 1131-3.

Les fonctions élémentaires, les blocs fonctions élémentaires (EFB) et les blocs fonctions dérivés (DFB) sont assemblés avec des signaux (variables) dans les sections FDB en un plan de fonctionnement. La taille d'une section FDB est de 23 lignes et 30 colonnes.

Les EFB sont munis d'un nombre variable ou invariable de variables d'entrée et peuvent être placés au choix sur les sections. Variable et EFB peuvent être commentés individuellement, les commandes de déclenchement sur une section peuvent être commentées par zones de texte à des endroits quelconques. L'exécution de tous les EFB peut être conditionnelle ou inconditionnelle.

Pour faciliter leurs recherches, tous les EFB sont regroupés en bibliothèques suivant leurs domaines d'applications et leurs fonction réparties sur différents groupes.

Editeur LD

L'Editeur LD (voir *Plan de contact LD, p. 237*) permet la programmation graphique du plan de contact conformément à la norme CEI 1131-3.

Contacts et bobines sont assemblés avec des signaux (variables) dans les sections LD en un plan de contact.

La taille d'une section FDB est de 23 lignes et 30 colonnes.

En outre, les fonctions et blocs fonctions élémentaires (EFB) évoqués dans l'éditeur FDB, les blocs fonctions dérivés (DFB) et les blocs définis par l'utilisateur (UDFB) sont également intégrables dans le plan de contact (voir *Editeur FDB, p. 16*).

La structure d'une section LD correspond à une voie de courant pour des montages à relais. Sur son côté gauche, se trouve le rail conducteur gauche. Cette barre d'alimentation gauche correspond à la phase (conducteur L) d'une voie de courant. De même que sur une voie de courant, pour la programmation LD ne sont "traités" que les objets LD (contacts, bobines) qui sont branchés sur l'alimentation en courant, c'est-à-dire qui sont reliés à la barre gauche d'alimentation. La barre droite d'alimentation, qui correspond au conducteur neutre, n'est pas représentée de manière optique. Toutefois, toutes les bobines et sorties d'EFB sont connectées à celui-ci à l'intérieur du montage, créant ainsi une continuité de courant.

Editeur SFC

L'Editeur SFC (voir *Langage de commande séquentielle SFC, p. 273*) permet la programmation graphique de la commande séquentielle conformément à la norme CEI 1131-3.

Les objets SFC sont assemblés sur une section SFC en une commande séquentielle conformement à la tâche à exécuter. La taille d'une section SFC est de 32 colonnes et 200 lignes.

Les objets suivants sont disponibles dans Concept pour la programmation d'une commande séquentielle :

- Etape (avec actions et section d'action)
- Transition (avec section de transition)
- Dérivation et jonction (Fan-In) alternatifs
- Dérivation et jonction (Fan-In) parallèles
- Saut
- Connexion

Des fonctions de contrôle sont déjà intégrées dans les étapes qui permettent un diagnostic facile.

Editeur IL

L'Editeur LD (voir *Liste d'instructions IL, p. 325*) permet la programmation sous forme de liste d'instructions conforme à la norme CEI 1131-3.

Des instructions IL composées d'opérateurs (commandes) et d'opérandes (signaux, variables), des fonctions et blocs fonctions élémentaires et des blocs fonctions dérivés sont écrits de manière consécutive sous format de texte sur les sections IL.

Vous pouvez utiliser toutes les fonctions standards disponibles sur Windows pour l'introduction de programme ainsi que quelques commandes supplémentaires pour le traitement de texte. La taille maximale d'une section IL est de 64 Ko.

Les opérateurs suivants sont disponibles dans Concept pour la programmation d'une liste d'instructions :

- Les instructions logiques (AND, OR, etc.)
- Les instructions arithmétiques (ADD, SUB, MUL, DIV, ...)
- Les instructions de comparaison (EQ, GT, LT, ...)
- Echelons (JMP, ... conditionnel/inconditionnel)
- Lancement d'EFB (CAL ... conditionnel/inconditionnel)

IL est programmé en format texte. Vous pouvez utiliser toutes les fonctions standards disponibles sous Windows pour le traitement de texte lors de saisie du texte. L'éditeur IL contient quelques commandes supplémentaires pour le traitement de texte.

Le saisie des textes (instructions, mots clés, séparateurs) est accompagné du contrôle de syntaxe qui marque les erreurs par un encadrement en couleur.

Editeur ST

L'Editeur ST (voir *Texte structuré ST, p. 399*) permet la programmation sous forme de texte structuré conforme à la norme CEI 1131-3.

Des déclarations ST (Statements) composées d'expressions (liste d'opérateurs) et d'opérandes (signaux, variables), des fonctions et blocs fonctions élémentaires et des blocs fonctions dérivés sont écrits sous format de texte sur les sections ST.

Vous pouvez utiliser toutes les fonctions standards disponibles sur Windows pour l'introduction de programme ainsi que quelques commandes supplémentaires pour le traitement de texte. La taille maximale d'une section ST est de 64 Ko.

Les déclarations et opérateurs suivants sont disponibles dans Concept pour la programmation sous format de texte structuré :

- Exécution conditionnelle/inconditionnelle des déclarations (IF, ELSIF, ELSE, ...)
- Exécution conditionnelle/inconditionnelle des boucles (WHILE, REPEAT)
- Opérateurs mathématiques, de comparaison et logiques
- Lancement conditionnel/inconditionnel d'EFB

ST est programmé en format texte. Vous pouvez utiliser toutes les fonctions standards disponibles sous Windows pour le traitement de texte lors de saisie du texte. L'éditeur ST contient quelques commandes supplémentaires pour le traitement de texte.

Le saisie des textes (instructions, mots clés, séparateurs) est accompagné du contrôle de syntaxe qui marque les erreurs par un encadrement en couleur.

Editeur LL984

Instructions, contacts, bobines et signaux (variables) sont assemblés en un LD (diagramme Ladder) par l'Editeur LL984 (voir *Ladder Logic 984, p. 463*) similaire à Modsoft. Instructions, contacts, bobines et variables peuvent être commentés.

La structure d'une section LL984 correspond à un trajet de courant dans les montages en relais. Sur son côté gauche, se trouve le rail conducteur gauche qui n'est cependant pas représenté. Cette barre d'alimentation gauche correspond à la phase (conducteur L) d'une voie de courant. A l'instar d'un trajet de courant, lors de la programmation avec LL984, seuls les objets LL984 (contacts, bobines) connectés à un réseau d'alimentation, c'est-à-dire au rail conducteur gauche sont "traités". Le rail conducteur de droite qui correspond au conducteur neutre n'est pas illustré non plus. Toutefois, toutes les bobines et sorties d'instructions sont connectées à celui-ci à l'intérieur du montage, créant ainsi une continuité de courant.

Pour la programmation d'un plan de contact avec LL984, Concept contient diverses instructions déjà prédéfinies. Vous les trouvez dans la bibliothèque des blocs LL984. D'autres instructions pour des applications spéciales sont disponibles sous forme d'instructions chargeables et peuvent être chargées.

Editeur de variables

Tous les noms de signaux (variables) symboliques nécessaires sont déclarés et commentés par l'Editeur de variables (voir *Editeur de variables, p. 559*). Seules les variables déclarées peuvent être utilisées dans les programmes Concept.

A chaque nom de signal symbolique doit être affecté un type de données ! Si une adresse de référence est affectée à ces variables, on obtient une Variable non-localisée (sans adresse de référence = Variable non-localisée). Chaque variable peut être dotée d'une valeur initiale supplémentaire qui est transférée dans l'API lors du premier chargement.

Editeur de type de données (Editeur DDT)

L'Editeur de type de données (voir *Types de données dérivés, p. 581*) vous permet de définir vos propres Types de données dérivés (Derived Data Type = DDT).

Les types de données dérivés réunissent plusieurs types de données élémentaires (BOOL, WORD, ...) en un article. Peuvent être réunis uniquement les mêmes types de données en ARRAY tout comme les types de données différents en STRUCT. Un certain nombre de types de données dérivés est déjà disponible dans Concept que vous pouvez par exemple utiliser pour les DFB.

Sur les DFB ou les EFB, les DDT sont représentés comme une seule connexion, c'est-à-dire qu'il n'est par exemple nécessaire de définir qu'une seule variable dans FBD. Il est donc conseillé de définir les groupes de types de données élémentaires récurrents (ainsi que les DDT) comme DDT afin d'améliorer ainsi la clarté de l'application concernée.

La définition se fait sous forme de texte, et vous pouvez utiliser pour ce faire toutes les fonctions standards disponibles sur Windows ainsi que quelques commandes supplémentaires pour le traitement de texte. La taille maximale d'un fichier type de données est de 64 Ko.

Editeur des données de référence

L'Editeur des données de référence (voir *Editeur des données de référence, p. 611*) vous permet d'afficher la valeur des variables en mode connecté, de forcer et de fixer des variables. Vous avez en outre la possibilité de séparer les variables du processus. Les entrées peuvent être sauvegardées dans un fichier et réutilisées ultérieurement.

Fonctions en ligne

Fonctions en ligne disponibles

Après la connexion du terminal avec l'API, tout un ensemble de fonctions en ligne sont disponibles pour la mise en marche et la maintenance :

- le programme installé sur le terminal est comparé avec le programme installé sur l'API
- l'API peut être démarré et arrêté
- des informations objets sont affichées
- des programmes peuvent être chargés, des sections peuvent être modifiées en ligne et chargées
- les valeurs de variables peuvent être définies en ligne
- le mode Animation affiche le programme avec les états courants

Commande et contrôle

Il n'est pas nécessaire de déclarer des variables spéciales de commande et de contrôle dans Concept. Les variables à visualiser peuvent être identifiées comme telles dans l'éditeur de variables, et puis exportées dans un fichier de configuration ModLink ou FactoryLink. Ce fichier peut être utilisé pour la visualisation.

Communication

Description

La préparation de la communication entre l'API et un autre abonné Modbus, Modbus Plus, SY/MAX-Ethernet ou TCIP/IP-Ethernet est effectuée, pour les langages CEI (FBD, LD, SFC, ST, IL), avec les EFB de la bibliothèque de blocs COMM. Pour le langage de programmation LL984, l'instruction MSTR est mise à disposition pour l'établissement de ces communications.

Indépendamment de ces blocs/instructions, il est également possible de procéder à un transfert Peer to Peer des contenus des registres avec Peer Cop.

La préparation de la communication entre l'API et une E/S décentralisée via Interbus est effectuée par simple inscription du module NOA sur la liste des affectations et chargement d'une instruction chargeable (ULEX).

La préparation de la communication entre le terminal et un API via Ethernet est effectuée par simple inscription et paramétrage des modules de couplage correspondants sur la liste des affectations.

Application protégée

Introduction

Dans certains domaines de l'industrie, pour des raisons de sécurité, une réglementation de l'accès à l'API l'enregistrement des modifications apportées au cours de l'exécution du programme et un archivage de ces protocoles de manière à ce qu'ils ne puissent pas être falsifiés sont une nécessité. Pour répondre à ces exigences, Concept a été doté de nouvelles fonctions dont l'utilisation permet d'obtenir une application protégée. Pour être sûr que tous les paramètres ont été définis, l'utilisateur peut cocher la case **Application sécurisée** de la boîte de dialogue **Projet** → **Propriétés du projet**. Cela permet à Concept de vérifier si tous ces paramètres ont été définis et qu'ils contiennent des valeurs valides. A la suite de cela, le projet sera muni du code prévu pour les applications protégées et fera partie des informations transmises lors du chargement dans l'API.

Application sécurisée

Dans la boîte de dialogue **Projet** → **Propriétés du projet**, le fait de cocher la case **Application sécurisée** exécute l'application protégée. Ce paramètre est également importé, exporté, lu et chargé dans l'API.

Note : Lors de l'activation de l'application protégée, l'état passe sur DIFFERENTS, ce qui nécessite un nouveau chargement dans l'API. Lorsque vous désactivez cette case à cocher, l'état passe à DIFFERENTS, ce qui rend dans ce cas également un nouveau chargement nécessaire. Cependant, lors de la connexion de Concept à un API ayant déjà chargé le paramètre "Application protégée", ce paramètre est appliqué automatiquement lors de la lecture.

Le fichier journal est consigné dans le répertoire Concept et son nom correspond à la date actuelle (ANNEEMOISJOUR.ENC, p. ex. 20020723.ENC). Dans la boîte de dialogue **Environnement commun**, vous pouvez entrer le chemin d'accès au fichier journal. Si vous n'entrez pas de chemin d'accès, Concept reprend le chemin par défaut du fichier journal (à savoir le répertoire Concept, p. ex. C:\CONCEPT).

La journalisation de l'accès en écriture à l'API permet d'enregistrer, entre autres, les informations suivantes :

- Nom de section
- Nom d'instance EFB/DFB, nom de type FB
- Nom de broche
- [Nom de variable] [Libellé] [Adresse]
- Ancienne valeur
- Nouvelle valeur
- Nom d'utilisateur (si le mot de passe (annonce) Concept a été activé dans Sécurité Concept)
- Date et heure (voir aussi *Format d'adresse dans le fichier journal [Logging], p. 1159*)

Condition

L'application protégée ne peut être exécutée que si les conditions suivantes sont remplies :

- uniquement utilisable pour 140 UC 434 12A ou 140 UC 534 14A/B,
- au moins une section CEI (si une telle section n'est pas disponible, le chargement est interrompu.).
- Mode Hors ligne (**En ligne** → **Déconnecter...**)
- Droits de superviseur (voir dans Concept sous **Aide** → **A propos de...** → **Utilisateur actuel :**)

Combinaisons permettant d'activer l'application protégée

On distingue diverses combinaisons d'activation de l'application protégée :

la case "Application protégée" à été cochée dans Concept,	**l'"Application protégée" a été chargée dans l'API.**	**Réaction lors d'une connexion à l'API**
Désactivée	Désactivée	Mode normal sans application protégée.
Désactivée	Activée	Lors de la lecture, la case **Application sécurisée** est cochée dans Concept et le chiffrement des fichiers journaux est activée.
Activée	Désactivée	Chargement nécessaire, car l'état est sur DIFFERENTS.
Activée	Activée	Mode normal avec application protégée (p. ex. chiffrement du fichier journal).

Lecture du fichier journal chiffré

Pour lire le fichier journal chiffré, l'outil Afffichage est exécuté automatiquement dans la fenêtre **Afficher journal**.

Note : Si un fichier journal a été modifié bien que cela soit illicite, le système décode ce fichier dans la mesure du possible et les lignes falsifiées restent illisibles. De plus, le message : "Ce fichier journal a été falsifié." apparaît dans la première ligne.

Programmes utilitaires

Introduction

En plus de Concept, tout un ensemble de programmes utilitaires est à votre disposition :

- Concept-DFB
- Concept-EFB
- Concept-SIM (16 bits)
- Concept-PLCSIM32 (32 bits)
- Concept-Security
- Concept-WinLoader
- Convertisseur Concept
- Concept-ModConnect

Concept-DFB

Concept-DFB sert à créer des DFB (Derived Function Blocs) (voir *DFB (blocs fonctions dérivés), p. 491*) et des Macros (voir *Macros, p. 535*).

DFB

Les DFB peuvent être utilisés aussi bien pour la structuration que pour la hiérarchisation d'un programme. D'un point de vue technique de programme, un DFB représente un sous-programme.

Les DFB peuvent être créés dans les langages de programmation FDB, LD, IL, ST. Dans Concept, les DFB peuvent être lancés par toutes les langages de programmation (FDB, LD, IL, ST) quelle que soit le langage de programmation dont on se sera servi pour leur création. Dans un DFB, un ou plusieurs DFB déjà existants peuvent être lancés, et les DFB ainsi lancés peuvent eux-mêmes en lancer un ou plusieurs autres.

Macros

Les macros servent à dupliquer les sections très souvent utilisées et les réseaux (y compris leurs logiques, leurs variables et leur déclaration de variables).

Les macros ont les propriétés suivantes :

- Les macros peuvent être créés uniquement dans le langage de programmation FDB.
- Les macros ne contiennent qu' une seule section.
- Les macros peuvent contenir une section d'une complexité quelconque.
- Du point de vue technique de programme, une macro instanciée, c'est-à-dire une macro insérée dans une section, ne se distingue pas d'une section conventionnelle.
- Il est possible de lancer des DFB dans une macro.
- Il est possible de déclarer pour la macro des variables propres à la macro.
- Il est possible d'utiliser des structures de données propres à la macro
- Application automatique des variables déclarées dans la macro.
- Il est possible de définir des valeurs initiales pour les variables macro.
- Il est possible d'instancier plusieurs fois une macro dans le programme d'ensemble avec des variables différentes.
- Les noms des sections, des variables et des structures de données peuvent comporter le symbole ~ comme caractère de remplacement.

Concept-EFB

L'outil en option Concept-EFB vous permet de créer des Fonctions et des Blocs fonctions (EFB) spécifiques de l'usager dans le langage évolué C++ et de les intégrer sous forme de groupes de bibliothèques dans votre version de Concept.

Pour le travail avec ces blocs définis par vous (UDFB) les mêmes règles que celles pour les EFB standards s'appliquent.

Il est par exemple conseillé de créer avec Concept-EFB les parties de programme complexe ayant un nombre élevé d'appels et les parties de programme dont l'utilisateur ne doit pas découvrir la solution, tels que les objets de technologie etc.

Note : Concept-EFB n'est pas livré avec Concept, il peut être commandé à part.

Concept-SIM (16 bits)

Pour simuler un API, c'est-à-dire tester votre programme utilisateur en ligne sans matériel, vous pouvez utiliser le simulateur 16 bits Concept-SIM (voir *Simuler un API (Simulateur 16 bits), p. 775*). Concept-SIM simule un API connecté via Modbus Plus.

Note : Le simulateur est disponible uniquement pour les langages CEI (FDB, SFC, LD, IL et ST).

Concept-PLCSIM (32 bits)

Pour simuler un API, c'est-à-dire tester votre programme utilisateur en ligne sans matériel, vous pouvez utiliser le simulateur 32 bits Concept-PLCSIM32 (voir *Simuler un API (Simulateur 32 bits), p. 777*). Concept-PLCSIM32 simule un API connecté via TCP/IP, et il également est possible de simuler les états des modules d'E/S. Jusqu'à cinq terminaux peuvent être connectés au même moment à l'API simulé.

Note : Le simulateur est disponible uniquement pour les langages CEI (FDB, SFC, LD, IL et ST).

Concept-Security

Concept-Security (voir *Concept-Security, p. 787*) vous permet d'octroyer des autorisations d'accès. Une autorisation d'accès entraîne une restriction des fonctions de Concept et de ses programmes utilitaires aux utilisateurs autorisés.

L'autorisation d'accès définie pour un utilisateur est valable pour tous les projets de l'installation Concept. 128 utilisateurs au maximum peuvent être définis.

Convertisseur Concept	Projets, DFB, macros, structures de données (types de données dérivés), créés dans une ancienne version de Concept peuvent être copiés sans problème et sans gros efforts dans la nouvelle version de Concept avec le convertisseur Concept (voir *Convertir Projets/DFB/Macros, p. 1023*).
Concept-EXECLoader	Concept-EXECLoader vous permet de charger les fichiers Exec dans l'API.
Concept-ModConnect	Concept-ModConnect (voir *Concept ModConnect, p. 1027*) vous permet d'ajouter de nouveaux modules (vos propres modules) d'E/S au configurateur.

Nouveautés de Concept 2.6 par rapport à Concept 2.5

2

Introduction

Vue d'ensemble

Ce chapitre décrit les nouvelles possibilités qu'offre Concept 2.6 par rapport à Concept 2.5.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Nouvelles fonctionnalités de Concept 2.6 par rapport à Concept 2.5	30
Nouvelles fonctionnalités de Concept 2.6 SR2 par rapport à Concept 2.6 SR1	36
Nouvelles fonctionnalités de Concept 2.6 SR3 par rapport à Concept 2.6 SR2	39

Nouvelles fonctionnalités de Concept 2.6 par rapport à Concept 2.5

Points importants

Nouvelles fonctionnalités générales :

- **Sections d'interruption**
- **Variables globales**
- **Fonctionnalités de sécurité**

Nouveaux EFB

Nouveaux EFB dans la bibliothèque SYSTEME :

Nouveaux EFB	Description
I_LOCK	Désactivation de toutes les sections d'interruption
I_UNLOCK	Activation de toutes les sections d'interruption
I_MOVE	Affectation protégée par interruption
ISECT_OFF	Désactivation de certaines sections d'interruption
ISECT_ON	Activation d'une certaine section d'interruption
ISECT_STAT	Etat de section d'interruption
PRJ_VERS	Nom et version de projet
GET_IEC_INF	Lecture de drapeaux d'état CEI
RES_IEC_INF	Réinitialisation des drapeaux d'état CEI

Nouveaux EFB dans la bibliothèque COMM :

Nouveaux EFB	Description
PORTSTAT	Etat du port Modbus

Démarrer Concept

Nouvelles fonctionnalités au démarrage de Concept :

Nouvelles fonctionnalités	Description
Connexion automatique à un API **quelconque**	Lors d'un démarrage via le symbole de projet Concept, une connexion automatique est établie à un API quelconque. Cette connexion est définie via les paramètres de ligne de commande (voir *Connexion automatique via des paramètres de ligne de commande (Modbus, Modbus +, TCP/IP), p. 1198*).
Lors d'un démarrage de Concept via l'outil CCLaunch, une connexion est établie automatiquement à un API **quelconque**.	Pour les grands réseaux, un fichier topologique est d'abord créé, puis utilisé dans l'outil CCLaunch. Cela vous permet alors de créer un itinéraire complet de routage MB+ (voir *Connexion automatique via l'outil CCLaunch (Modbus Plus), p. 1202*) établissant une connexion automatique à l'API.
Affichage des derniers projets/ DFB ouverts	Au démarrage de Concept, une liste des derniers projets/DFB (4 au total) ouverts apparaît dans le menu principal **Fichier**.
Affichage du contenu d'archive	Lors de la décompression d'un projet archivé, tous les fichiers de l'archive sont préalablement affichés.

Animation

12 jeux de couleurs différents à des fins d'animation dans l'éditeur FBD, IL, ST, SFC et LD :

Nouvelle fonctionnalité	Description

CONCEPT.INI:
`[Colors]`
`AnimationColors= (0-12)`
	Définit le jeu de couleurs d'une animation en ligne dans tous les éditeurs.

Editeur des données de référence

Nouvelle fonctionnalité de l'éditeur de données de référence :

Nouvelle fonctionnalité	Description
Format d'adresse CEI (QW0000X)	Permet de visualiser le format d'adresse CEI (QW0000X).

Fonctions En ligne

Nouvelles fonctionnalités en ligne :

Nouvelles fonctionnalités	Description
Protection de Quantum par mot de passe	L'API Quantum est protégé en écriture par mot de passe.
Sections d'événements	Des diagnostics en ligne pour sections d'interruption sortent sur l'écran.
Affichage des événements	Dans un Fichier INI (voir *Configuration INI de l'affichage des événements [Online Events] (événements en ligne), p. 1165*) spécifique d'un projet, vous pouvez définir des descriptions d'erreurs devant apparaître dans l'Affichage des événements (**En ligne** → **Affichage événements**).

Boîte de message

Nouvelle option du menu contextuel :

Nouvelles fonctionnalités	Description
Enregistrement des messages	Après avoir été affichés, les messages peuvent être inscrits dans un fichier à l'aide de la commande **Sauvegarder messages...** (menu principal **Fenêtre**).

Nouvelle UC

Nouvelle UC :

Gamme d'automates	Description
Atrium	CPU 180-CCO-241-11

Nouveaux modules

Nouveaux modules Quantum :

Module	Description
140-NOE-771-01	Module Ethernet sans fonctionnalité de redondance d'UC.
140-NOE-771-11	Module Ethernet (Factory Cast) sans fonctionnalité de redondance d'UC.
140-CPS-114-20	Module d'alimentation
140-CPS-124-20	Module d'alimentation
140-NOG-111-00	Module maître 1/SFB
140-NWM-100 00	Module Ethernet (Factory Cast HMI)

Nouveau module Momentum :

Module	Description
170-ANR-120-91	Module d'E/S analogique/numérique

Navigateur de projet

Nouvelles fonctionnalités du navigateur de projet :

Nouvelles fonctionnalités	Description
Affichage de sections d'interruption	Lors de l'utilisation de sections d'événements d'E/S et de temporisation, celles-ci apparaissent dans l'arborescence du navigateur de projet.
Fenêtre détaillée	La fenêtre du navigateur se partage en deux volets verticaux et le second affiche les niveaux inférieurs (p. ex. les DFB inclus et les sections de transition, etc.) de l'élément sous forme d'arborescence.

Analyser section

Nouvelles fonctionnalités au niveau de l'analyse de sections :

Nouvelle fonctionnalité	Description
Analyse de sections d'interruption	Une analyse supplémentaire est exécutée pour les sections d'interruption.
Analyse de variables globales dans des DFB	Une analyse est exécutée pour les variables globales de DFB.

DFB

Nouvelles fonctionnalités au niveau de la programmation de DFB :

Nouvelle fonctionnalité	Description
Variables localisées	Les variables localisées sont autorisées dans les DFB, dans la mesure où l'option de la boîte de dialogue **Extensions CEI** a été activée. Des variables localisées dans des DFB permettent de créer des variables globales pour l'ensemble du programme.

Types de données

Nouvelles fonctionnalités au niveau de la programmation de DFB :

Nouvelle fonctionnalité	Description
Visualisation des commentaires pour les éléments de la structure de données	Le système affiche les commentaires pour types de données de composants, définis dans des fichiers de types de données (*.ddt, *.dty) : ● la ligne d'état des éditeurs ● l'éditeur de variables dans le cadre de la définition de valeurs initiales ● la zone Inspect de l'animation
Définition avancée de type de données (supérieure à 64 Ko), p. 589	Pour la définition de types de données locale, la limite de 64 Ko n'est plus applicable lors de l'introduction de fichiers Include non localisés.

Configuration

Nouvelles fonctionnalités du Configurateur :

Nouvelle fonctionnalité	Description
Configuration du coupleur 1/SFB	Permet de gérer les modules d'E/S A500/A350. Plage d'E/S étendue à 160 mots d'entrée/sortie.
Paramètres de sécurité Quantum	Dans la nouvelle boîte de dialogue (sous-menu d'**Extensions de Config**), vous pouvez définir les paramètres suivants : ● Zone de données sûre ● Restriction d'écriture réseau ● Activation de l'option de déconnexion automatique
Configuration d'Interbus avec Atrium	La configuration d'Interbus se fait avec les UC 180 CCO 241 01 (= 1 INTERBUS) et 180 CCO 241 11 (= 2 INTERBUS) Atrium.

Journalisation (*.LOG, *.ENC)

Nouvelles fonctionnalités dans le cadre de la journalisation :

Nouvelle fonctionnalité	Description
Contenu supplémentaire	Lors de la consigne dans le fichier journal de la protection en écriture de l'API, le système affiche également les modifications apportées aux valeurs des variables et des libellés.
Nouveau format Date/Heure	L'activation de la case à cocher **Format de date universel** dans la boîte de dialogue **Environnement commun** (ce paramètre influe aussi sur le fichier CONCEPT.INI) permet de modifier le format par défaut. A la suite de cela, le mois apparaît dans Concept, en trois caractères et en anglais. **Exemple :** 24-Dec-2002 14:46:24
Chiffrement du protocole	L'activation de la case à cocher **Chiffrement du journal** dans la boîte de dialogue **Environnement commun** (ou indirectement à l'aide de la case à cocher **Application sécurisée** dans la boîte de dialogue **Propriétés du projet**) permet de chiffrer la journalisation de l'accès en écriture à l'API. Le nom du fichier codé a l'extension *.ENC.

Application protégée

Nouvelles fonctionnalités pour une application protégée :

Nouvelle fonctionnalité	Description
Protéger l'application	L'activation de cette case à cocher de la boîte de dialogue **Projet** → **Propriétés du projet** permet au système de coder et de consigner les modifications du programme dans un fichier journal *.ENC. Exportation/importation permet de lire ce paramètre et de le transmettre à l'API.

Nouveaux outils

Nouveaux outils de Concept :

Nouvel outil	Description
CCLaunch	Cet outil permet d'établir une connexion automatique (voir *Connexion automatique via l'outil CCLaunch (Modbus Plus), p. 1202*) à un API au sein d'un grand réseau.
Outil d'affichage	Cet outil permet de visualiser des protocoles chiffrés (*.ENC). La commande de menu **Afficher journal** permet de l'exécuter automatiquemeent lorsque le chiffrement de protocole a été activé.

Nouvelles fonctionnalités de Concept 2.6 SR2 par rapport à Concept 2.6 SR1

Nouveaux EFB

Nouveaux EFB dans la bibliothèque CEI :

Nouveaux EFB	Description
CMPR	Compare la configuration binaire de la matrice A à celle de la matrice B.
MBIT avec pointeur	Modifie la position des bits dans une matrice de données.
SEARCH	Recherche une configuration binaire spécifique dans les registres d'un tableau source.
SENS avec pointeur	Vérifie la valeur de scrutation d'une position de bits spécifique dans une matrice de données.
XXOR	Exécute une opération booléenne en OU exclusif avec les configurations binaires des matrices source et cible.

Recherche/ remplacement de FFB

Nouvelles fonctionnalités de recherche/remplacement de FFB :

Nouvelle fonctionnalité	Description
Le type de FFB est remplacé dans toutes les sections (DFB uniquement)	Dans la boîte de dialogue **Remplacer type FFB**, vous pouvez demander à remplacer le type FFB sélectionné dans toutes les sections (DFB uniquement) en cochant la nouvelle case disponible **Remplacer dans toutes les sections**.

Création d'un nouveau projet

Nouvelles fonctionnalités pour la création d'un nouveau projet :

Nouvelle fonctionnalité	Description
Définition d'un chemin d'accès au projet lors de la création d'un nouveau projet	Lorsque vous créez un nouveau projet (**Fichier** → **Nouveau projet**), vous pouvez désormais définir un chemin d'accès personnalisé ou toujours reprendre le chemin d'accès proposé par défaut.

Nouvelles options de la boîte de dialogue de lecture et de chargement

Nouvelles options de la boîte de dialogue de lecture et de chargement :

Nouvelles fonctionnalités	Description
Nouvelles cases à cocher dans la boîte de dialogue **Charger dans l'automate** : • Mémoire d'état + valeurs initiales • Mémoire d'état uniquement	Si vous cochez la case **Mémoire d'état + valeurs initiales**, le système copie toutes les valeurs initiales des variables 4x localisées de l'éditeur de variables dans le miroir de la mémoire d'état. Il charge ensuite dans l'automate les valeurs initiales ainsi que tous les bits d'E/S 0x et 1x bloqués du miroir de la mémoire d'état. Si vous cochez la case **Mémoire d'état uniquement**, le système charge dans l'automate les valeurs initiales des variables 4x localisées et tous les bits d'E/S 0x et 1x bloqués du miroir de la mémoire d'état.
Nouvelles cases à cocher dans la boîte de dialogue **Lire API** : • Lire mémoire d'état + valeurs initiales • Lire uniquement la mémoire d'état	Si vous cochez la case **Lire mémoire d'état + valeurs initiales**, le système lit en premier lieu toutes les valeurs 0x, 1x et 4x localisées de l'automate et les enregistre dans le miroir de la mémoire d'état. Il remplace ensuite les valeurs initiales des variables 4x par la valeur figurant dans le miroir de la mémoire d'état. Si vous cochez la case **Lire uniquement la mémoire d'état**, le système lit toutes les valeurs 0x, 1x et 4x localisées de l'automate et les enregistre dans le miroir de la mémoire d'état.

Fichiers INI

Nouveaux paramètres dans le fichier CONCEPT.INI :

Nouveaux paramètres	Description
Définition de l'écrasement des valeurs lues de la mémoire d'état	Sur la ligne [RDE] du fichier CONCEPT.INI, vous pouvez spécifier que les valeurs lues de la mémoire d'état ne doivent pas être écrasées dans l'éditeur de données par des opérations en ligne.
Définition du lancement de l'animation de l'éditeur de données	Sur la ligne [RDE] du fichier CONCEPT.INI, vous pouvez spécifier que l'animation de l'éditeur de données doit être lancée automatiquement à l'ouverture d'un tableau.
Exclusion de tous les DFB ou des DFB globaux de l'enregistrement en ligne	Sur la ligne [Backup] du fichier CONCEPT.INI, vous pouvez spécifier que les répertoires "DFB" et/ou "DFB.GLB" ne doivent pas se trouver dans le répertoire de sauvegarde à l'issue de l'enregistrement en ligne.

Nouveaux paramètres dans le fichier NomDuProjet.INI :

Nouveau paramètre	Description
Définition du chemin d'accès et des fichiers de sauvegarde	Sur la ligne [Backup] du fichier NomDuProjet.INI, vous pouvez éditer un fichier batch (fichier EXE) pour l'opération d'enregistrement en ligne avec lequel vous pouvez effectuer des sauvegardes supplémentaires, par exemple vers un autre ordinateur.

Affectation multiple des adresses

Nouvelle fonctionnalité en cas d'affectation multiple d'adresses :

Nouvelle fonctionnalité	Description
Elimination de l'affectation multiple d'une même adresse due à des noms de variables différents	La boîte de dialogue **Affectation multiple des adresses** permet de remplacer ou de renommer des variables toutes affectées à la même adresse. A l'issue de cette opération, seul un nom de variable est affecté à l'adresse en question.

Nouvelles fonctionnalités de Concept 2.6 SR3 par rapport à Concept 2.6 SR2

Nouveaux commande

Nouveaux commande:

Nouveaux commande	Description
Options → Tools	Cette commande vous permet d'ouvrir un menu qui peut être utilisé pour exécuter des applications ou programmes d'aide supplémentaires.

Structure des projets

3

Introduction

Aperçu

Ce chapitre décrit la structure des projets dans Concept.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Structure et traitement des projets	42
Programme	49
Sections	54
Données de configuration	60

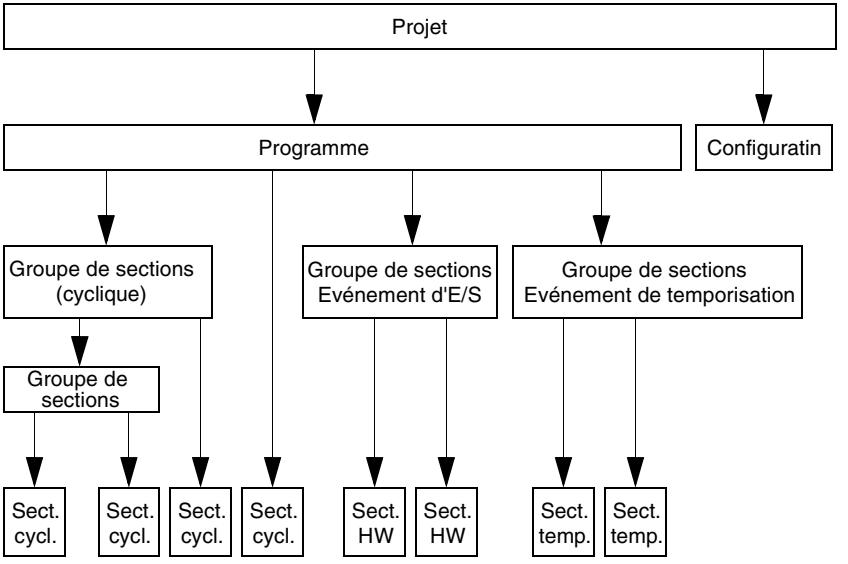
Structure et traitement des projets

Structure d'un projet

La création d'un programme d'API avec Concept se fait suivant une structure hiérarchique comportant un projet avec la Configuration API (voir *Données de configuration, p. 60*) et le Programme (voir *Programme, p. 49*). Le programme est subdivisé en groupes de sections et Sections (voir *Sections, p. 54*).

Au sein d'un projet, la configuration API et les éléments de programme nécessaires peuvent être créés dans un ordre quelconque (top down c'est-à-dire descendant ou bottom up c'est-à-dire ascendant).

Structure d'un projet :



Traitement d'un projet CEI/LL984

Le tableau décrit le traitement d'un projet LL984/CEI (Quantum) :

Echelon	Processeur logique	Processeur d'E/S
1	Gestion système (Overhead), par exemple communication avec NOM, NOE, etc.	-
2	Exécution du segment 1 LL984	Ecriture des sorties calculées dans segment n.
		Lecture des entrées nécessaires pour segment 2
3	Exécution du segment 2 LL984	Ecriture des sorties calculées dans segment 1.
		Lecture des entrées nécessaires pour segment 3
4	Exécution du segment 3 LL984	Ecriture des sorties calculées dans segment 2.
		Lecture des entrées nécessaires pour segment 4
...	...	...
n	Exécution du segment n (n =< 32) LL984	Ecriture des sorties calculées dans segment n-1.
		Lecture des entrées nécessaires pour segment 1
n+1	Exécution de la section 1 CEI	-
n+2	Exécution de la section 2 CEI	-
n+3	Exécution de la section 3 CEI	-
	..	-
m	Exécution de la section n CEI (n =< 1600) et retour à l'étape 1	-

1 La gestion système (par exemple communication avec les modules des liens NOM, NOE) a lieu à cette étape.

2 - 4 Dans ces étapes, le processeur logique exécute la logique des sections LL984 dans les segments 1-3 (en fonction des paramètres définis dans l'Ordonnanceur de segments (voir *Ordonnanceur de segments, p. 109*)).
Le processeur d'E/S transmet au matériel les valeurs de sortie calculées dans le précédent segment et le matériel lit simultanément les valeurs d'entrée nécessaires pour le segment suivant correspondant.

n Dans cette étape, le processeur logique exécute la logique des sections LL984 dans segment n.
Le processeur d'E/S transmet au matériel les valeurs de sortie calculées dans le précédent segment et le matériel lit simultanément les valeurs d'entrée nécessaires pour le segment 1.
Note : Les valeurs de sortie calculées dans ce segment ne seront exécutées que lors du prochain parcours de l'étape 2, c'est-à-dire après traitement de la partie logique CEI et de la gestion système. Vous devriez donc éviter d'exécuter une logique critique au niveau du temps dans ce segment.

n+1 - m Le processeur logique exécute la logique des sections CEI dans ces étapes.
Ensuite, a lieu le "Retour" à l'étape 1.
Note : Aucun signal matériel n'est lu ou écrit. Seules les valeurs calculées/lues dans les étapes 2 à n sont utilisées lors du traitement. Les valeurs de sortie calculées dans ces étapes sont transmises lors des étapes 2 à n (en fonction des paramètres définis dans l'ordonnanceur de segments).

Traitement d'un projet LL984

Le tableau décrit le traitement d'un projet LL984 (Quantum) :

Echelon	Processeur logique	Processeur d'E/S
1	Gestion système (Overhead), par exemple communication avec NOM, NOE, etc.	-
2	Exécution du segment 1 LL984	Ecriture des sorties calculées dans segment n.
		Lecture des entrées nécessaires pour segment 2
3	Exécution du segment 2 LL984	Ecriture des sorties calculées dans segment 1.
		Lecture des entrées nécessaires pour segment 3
4	Exécution du segment 3 LL984	Ecriture des sorties calculées dans segment 2.
		Lecture des entrées nécessaires pour segment 4
...	...	...
n	Exécution du segment n LL984 (n =< 32) et retour à l'étape 1	Ecriture des sorties calculées dans segment n-1.
		Lecture des entrées nécessaires pour segment 1

1 La gestion système (par exemple communication avec les modules des liens NOM, NOE) sont exécutés à cette étape.

2 - 4 Dans ces étapes, le processeur logique exécute la logique des sections LL984 dans les segments 1-3 (en fonction des paramètres définis dans l'Ordonnanceur de segments (voir *Ordonnanceur de segments, p. 109*)).
Le processeur d'E/S transmet au matériel les valeurs de sortie calculées dans le précédent segment et le matériel lit simultanément les valeurs d'entrée nécessaires pour le segment suivant correspondant.

n Dans cette étape, le processeur logique exécute la logique des sections LL984 dans segment n.
Le processeur d'E/S transmet au matériel les valeurs de sortie calculées dans le précédent segment et le matériel lit simultanément les valeurs d'entrée nécessaires pour le segment 1.
Ensuite, a lieu le "Retour" à l'étape 1.
Note : Les valeurs de sortie calculées dans ce segment ne seront exécutées que lors du prochain passage de l'étape 2, c'est-à-dire après traitement des frais généraux. Vous devriez donc éviter d'exécuter une logique critique au niveau du temps dans ce segment.

Traitement d'un projet CEI

Le tableau décrit le traitement d'un projet CEI (Quantum) :

Echelon	Processeur logique	Processeur d'E/S
1	Gestion système (Overhead), par exemple communication avec NOM, NOE, etc.	-
2	-	Ecriture des sorties affectées segment 1
		Lecture des entrées affectées segment 1
3	-	Ecriture des sorties affectées segment 2
		Lecture des entrées affectées segment 2
4	-	Ecriture des sorties affectées segment 3
		Lecture des entrées affectées segment 3
...	...	...
n	-	Ecriture des sorties affectées segment n (n =< 32)
		Lecture des entrées affectées segment n (n =< 32)
n+1	Exécution de la section 1 CEI	-
n+2	Exécution de la section 2 CEI	-
n+3	Exécution de la section 3 CEI	-
	..	-
m	Exécution de la section n CEI (n =< 1600) et retour à l'étape 1	-

1 La gestion système (par exemple communication avec les modules des liens NOM, NOE) a lieu à cette étape.

2 - n Dans ces étapes, le processeur d'E/S écrit et lit les signaux matériel des modules affectés aux segments correspondants (en fonction des paramètres définis dans l'Ordonnanceur de segments (voir *Ordonnanceur de segments, p. 109*)).

n+1 - m Le processeur logique exécute la logique des sections CEI dans ces étapes.
Ensuite, a lieu le "Retour" à l'étape 1.
Note : Aucun signal matériel n'est lu ou écrit. Seules les valeurs lues dans les étapes 2 à n sont utilisées lors du traitement. Les valeurs de sortie calculées dans ces étapes sont transmises lors des étapes 2 à n (en fonction des paramètres définis dans l'ordonnanceur de segments).

Traitement d'un projet CEI avec sections d'interruption

Le tableau décrit le traitement d'un projet CEI (Quantum) avec sections d'interruption :

Echelon	Processeur logique	Processeur d'E/S
1	Gestion système (Overhead), par exemple communication avec NOM, NOE, etc.	-
2	-	Ecriture des sorties affectées segment 1
		Lecture des entrées affectées segment 1
3	-	Ecriture des sorties affectées segment 2
		Lecture des entrées affectées segment 2
4	-	Ecriture des sorties affectées segment 3
		Lecture des entrées affectées segment 3
HE1	1. Section d'événements d'E/S, exécution spontanée, lors d'une interruption matérielle	-
HE2	2. Section d'événements d'E/S, exécution spontanée, lors d'une interruption matérielle	-
...	...	...
HE64	64. (dernière) section d'événements d'E/S, exécution spontanée, lors d'une interruption matérielle	-
TE1	1. Section d'événements de temporisation, exécution uniquement lors d'une interruption du temporisateur	-
TE2	2. Section d'événements de temporisation, exécution uniquement lors d'une interruption du temporisateur	-
...	...	...
TE16	16. Section d'événements de temporisation, exécution uniquement lors d'une interruption du temporisateur	-
...	...	...
n	-	Ecriture des sorties affectées segment n (n =< 32)
		Lecture des entrées affectées segment n (n =< 32)
n+1	Exécution de la section 1 CEI (cyclique)	-
n+2	Exécution de la section 2 CEI (cyclique)	-

Echelon	Processeur logique	Processeur d'E/S
n+3	Exécution de la section 3 CEI (cyclique)	-
	..	-
m	Exécution de la section n CEI (n =< 1600) et retour à l'étape 1	-

1 La gestion système (par exemple communication avec les modules des liens NOM, NOE) a lieu à cette étape.

2 - n Dans ces étapes, le processeur d'E/S écrit et lit les signaux matériel des modules affectés aux segments correspondants (en fonction des paramètres définis dans l'Ordonnanceur de segments (voir *Ordonnanceur de segments, p. 109*)).

n+1 - m Le processeur logique exécute la logique des sections CEI dans ces étapes.
Ensuite, a lieu le "Retour" à l'étape 1.
Note : Aucun signal matériel n'est lu ou écrit. Seules les valeurs lues dans les étapes 2 à n sont utilisées lors du traitement. Les valeurs de sortie calculées dans ces étapes sont transmises lors des étapes 2 à n (en fonction des paramètres définis dans l'ordonnanceur de segments).

HE1 - HE64 Si la valeur du signal matériel d'interruption attribué à une section donnée varie en fonction des paramètres définis, le traitement cyclique et le cas échéant, celui d'une section d'événements de temporisation en cours est immédiatement interrompu pour passer à la section d'événements d'E/S. En fin de traitement de toutes les sections d'événements (et d'événements de temporisation) à traiter, le système reprend le traitement cyclique à l'endroit où celui-ci avait été interrompu. (Voir également chapitre "*Sections d'interruption, p. 1190*")

TE1 - TE16 Si le signal d'interruption de temporisation défini pour une section donnée apparaît, le traitement cyclique est immédiatement interrompu pour passer à la section d'événements d'E/S. En fin de traitement des sections d'événements de temporisation à traiter, le système reprend le traitement cyclique à l'endroit où celui-ci avait été interrompu, dans la mesure où des sections d'événements de temporisation à traiter ne sont plus disponibles. (Voir également chapitre "*Sections d'événements de temporisation, p. 1172*")

Programme

Structure d'un programme

Un programme se compose d'une ou plusieurs sections (voir *Sections, p. 54*) ou groupes de sections. Des groupes de sections peuvent contenir des sections ou d'autres groupes de sections. Les groupes de sections peuvent uniquement être créés ou remplis à l'aide de **Projets** → **Navigateur de projet (voir *Navigateur de projet, p. 573*)**. Les sections décrivent le fonctionnement de l'intégralité du système.

Les variables, les constantes, les littéraux et les adresses directes utilisés sont également gérés dans le programme.

Variables : Les variables sont destinées à l'échange de données au sein d'une section, entre plusieurs sections et entre le programme et l'API.

La déclaration des variables s'effectue via la commande **Projet → Déclaration de variables**. Si une adresse est affectée à la variable par le biais de cette fonction, on parle alors de "variable localisée". Si aucune adresse n'est affectée à la variable, on parle de variable non localisée. Si un type de données est affecté à la variable, on parle de "variable multi-éléments".

Des constantes et des littéraux sont également disponibles.

Le tableau suivant fournit un aperçu des différents types de variables :

Type de variable	Description
Variables localisées	Une adresse de mémoire d'état (références 0x, 1x, 3x,4x) est attribuée aux variables localisées. La valeur de ces variables est enregistrée dans la mémoire d'état et peut être modifiée en ligne à l'aide de l'éditeur de données de référence. L'adressage de ces variables peut avoir lieu à l'aide de leur nom symbolique ou de leur adresse de référence. Toutes les entrées et les sorties de l'API sont reliées à la mémoire d'état. L'accès du programme aux signaux de périphériques connectés à l'API a lieu uniquement par le biais de variables localisées. Les accès de l'extérieur via interfaces Modbus ou Modbus Plus de l'API, tels que ceux de systèmes de visualisation, sont également possibles via les variables localisées.
Variables non localisées	Aucune adresse de mémoire d'état n'est attribuée aux variables non localisées. Elles n'occupent donc aucune adresse de mémoire d'état. La valeur de ces variables est enregistrée dans la mémoire d'état et peut être modifiée en ligne à l'aide de l'éditeur de données de référence. L'adressage de ces variables est possible uniquement par le biais du nom symbolique. Il est préférable de déclarer les signaux n'exigeant pas d'accès à la périphérie, tels que des résultats intermédiaires, des capteurs de système, etc., en tant que variables non localisées.
Variables multi-éléments	Variables auxquelles est affecté un type de données dérivé. Dans le cadre de ces variables, on distingue les variables structurées des variables de matrice.
Variables structurées	Variables auxquelles est affecté un type de données dérivé défini avec STRUCT (structure). Une structure est une collection d'éléments de données ayant généralement différents types de données (Types de données élémentaires et/ou Types de données dérivés).
Variables de matrice	Variables auxquelles est affecté un Type de données dérivé à l'aide du mot-clé ARRAY (matrice). Une matrice est une collection d'éléments de données ayant le même type de données.

Comportement de démarrage de variables

Dans le cadre du comportement au démarrage de l'API, on fait la différence entre le démarrage à chaud et le démarrage à froid :

- **Démarrage à froid**
  Après un démarrage à froid (chargement du programme avec **En ligne** → **Chargement**) toutes les variables et ceci indépendamment de leur type, sont mises sur "0" ou sur leurs valeurs initiales.
- **Démarrage à chaud**
  Lors d'un démarrage à chaud (arrêter et démarrer le programme ou **En ligne** → **Chargement des modifications**), on distingue différentes procédures de démarrage pour les variables localisées, les adresses directes et les variables non localisées :
  - **Variables localisées /adresses directes**
    Lors d'un démarrage à chaud, tous les registres 0x-, 1x- et 3x sont remis sur "0" ou sur leur valeur initiale (si elle est disponible).
    Les bobines mémorisées font ici exception. Elles conservent leur valeur actuelle (comportement de mémorisation).
    Les registres 4x conservent leur valeur actuelle (comportement de mémorisation).
  - **Variables non localisées**
    Lors d'un démarrage à chaud, toutes les variables non localisées conservent leur valeur actuelle (comportement de mémorisation).

Cette différence de comportement lors d'un démarrage à chaud conduit à des particularités de comportement au démarrage à chaud des fonctions Posit. et Réinit.

- **Posit. et Réinit. dans LD et IL**
  Le comportement de démarrage à chaud est fonction du type de variables utilisé (comportement mémorisant, en cas d'utilisation de variables non localisées; comportement non mémorisant en cas d'utilisation de variables/d'adresses directes).
- **Blocs fonctions SR et RS dans FDB, LD, IL et ST**
  Ces blocs fonctions utilisent une variable interne non localisée et ont donc toujours un comportement de mémorisation.

Constantes

Les constantes sont des variables non localisées auxquelles est affectée une valeur ne pouvant pas être modifiée par la logique de programme (lecture seule).

Littéraux (valeurs)

Les littéraux servent à écrire des entrées de FFB, des conditions de transition, etc. avec des valeurs directes. Ces valeurs ne peuvent pas être écrasées par la logique du programme (lecture seule).

Le valeurs de littéraux peuvent être modifiées en ligne.

Parmi les littéraux, on distingue les littéraux génériques et les littéraux classés dans un type.

Le tableau suivant fournit un aperçu des différents types de littéraux :

Littéral	**Description**
Littéraux génériques	Si le type de données du littéral n'est pas pertinent pour vous, indiquez simplement la valeur du littéral. Dans ce cas, Concept affecte automatiquement un type de données adapté au littéral.
Littéraux classés dans un type	Si vous voulez déterminer le type de données d'un littéral, vous pouvez le faire avec la construction suivante : "Nom du type de données"#"Valeur du littéral" par ex. INT#15 (type de données : Entier, valeur : 15), BYTE#00001111 (type de données : octet, valeur : 00001111) REAL#23.0 (type de données : réel, valeur : 23.0) Pour l'affectation du type de données **REAL**, il est également possible d'indiquer la valeur de la façon suivante : 23.0. Lorsque l'on indique une virgule décimale, le type de données REAL est automatiquement affecté.

Adresses directes

Les adresses directes sont des zones de mémoire sur l'API. Elles se trouvent dans la mémoire d'état et peuvent être affectées à des modules d'entrée/de sortie.

Les adresses directes peuvent être indiquées ou affichées dans différents formats. La définition du format d'affichage est exécutée dans la boîte de dialogue **Options** → **Environnement** → **Commun...**. La définition du format d'affichage n'a aucun impact sur le format d'entrée. Cela signifie que l'entrée d'adresses directes peut s'effectuer dans un format quelconque.

Les formats d'adresses suivants sont disponibles :

- **Format standard (400001)**
  L'adresse à cinq caractères est affichée immédiatement après le premier chiffre (la référence).
- **Format à séparateur (4 : 00001)**
  Le premier chiffre (la référence) est séparé par deux-points (:) de l'adresse à cinq caractères.
- **Format compact (4 : 1)**
  Le premier chiffre (la référence) est séparé par deux-points (:) de l'adresse suivante, les zéros en tête n'étant pas indiqués dans l'adresse.
- **Format CEI (QW1)**
  Un identificateur est placé en tête après CEI, suivi de l'adresse à cinq caractères.
  - %0x12345 = %Q12345
  - %1x12345 = %I12345
  - %3x12345 = %IW12345
  - %4x12345 = %QW12345

Les valeurs d'adresses directes peuvent être modifiées en ligne à l'aide de l'éditeur de données de référence (voir *Editeur des données de référence, p. 611*).

Comportement au démarrage de sorties TOR

Au démarrage de l'API, les sorties affectées à des registres 0x sont généralement supprimées. En revanche, les sorties TOR affectées à des registres 4x conservent la valeur actuelle, lors de l'arrêt ou du démarrage de l'API.

Sections

Introduction

Un programme se compose d'une ou plusieurs sections. Une section décrit le fonctionnement d'une unité technologique (d'un moteur, par exemple) d'un système.

Dans Concept, chaque section dispose de sa propre fenêtre de document. Pour des raisons de clarté, il est toutefois judicieux de diviser une section de taille importante en plusieurs petites sections. La barre de défilement permet de se déplacer dans la section.

Pour chaque section, les sauts de page peuvent être affichés de façon à pouvoir prendre en compte le format du papier lors de la programmation. Vous obtenez ainsi une impression lisible de la section.

Types de sections

Dans Concept, on distingue pour Quantum trois types de sections en fonction du traitement :

- Les **sections cycliques** sont traitées à chaque cycle programme. Le temps de réaction dépend du temps de cycle et correspond à au moins un cycle et deux cycles au plus.
- L'initialisation et le traitement des **sections d'événements d'E/S** ne sont pas cycliques, mais spontanées, dans la mesure où la valeur d'un signal d'interruption affecté donné varie (en fonction des paramètres définis dans le configurateur et les caractéristiques de la section).
  Le module 140-HLI-340-00 prévoit 16 entrées d'interruption. L'embase locale peut être équipée de 4 modules HLI au total.
  Le temps de réaction à l'événement d'E/S ne dépend en général que de la durée de traitement de blocs fonctions élémentaires à traiter sur la section concernée et du temps de mise au point.
- Les **sections d'événements de temporisation** sont initiées et traitées à intervalles précis définis par l'utilisateur.
  Ces intervalles sont définis par un multiple de 1 ms et une phase (décalage de cycle) dans la boîte de dialogue Caractéristiques de section d'événement de temporisation.
  Le temps de réaction dépend du temps de cycle. De cette manière, les réactions au niveau de sorties ont lieu à intervalles définis.

Nombre maximal de sections

Jusqu'à 1 600 sections sont possibles par programme.

Langages de programmation

Les sections peuvent être programmées dans les langages de programmation CEI suivants : FDB (langage de bloc fonction), LD (langage à contacts), SFC (automate d'enchaînement), IL (liste d'instructions) ou ST (littéral structuré) ou en LL984 (langage à contacts), langage comparable à Modsoft. Au sein d'une même section, seul un des langages de programmation mentionnés peut être utilisé.

Echange de valeurs

Au sein des sections, l'échange de valeurs s'effectue via des connexions, des variables ou des adresses directes. Entre les différentes sections, l'échange de valeurs s'effectue via des variables ou des adresses directes.

Ordre d'exécution des sections

Les sections LL984 sont exécutées en premier. L'ordre des sections LL984 peut être défini à l'aide de la boîte de dialogue **Projet** → **Configuration automate** → **Configurer** → **Ordonnanceur de segments...** . Une fois le traitement de toutes les sections LL984 terminé, le traitement des sections CEI (FDB, SFC, LD, IL, ST) est effectué. Vous pouvez déterminer leur ordre d'exécution à l'aide de la boîte de dialogue **Projet** → **Ordre d'exécution...** ou du Navigateur de projet (voir *Navigateur de projet, p. 573*).

Imprimer des sections

Lors de l'impression, les sections sont réparties en pages. La quantité d'informations contenue sur ces pages est fonction des paramètres définis dans **Fichier** → **Imprimer**. Pour afficher la répartition des pages, utilisez la commande **Affichage** → **Saut de page**.

Variable de section

Pour chaque section CEI (FDB, SFC, LD, IL et ST) une variable multi-éléments portant le même nom que la section est créée.

Cette variable est du type de données SECT_CTRL et elle comporte deux éléments :

- L'élément "disable" du type de données BOOL permet de désactiver des sections.
- L'élément "hsbyState" du type de données BYTE permet d'afficher l'état de redondance d'UC de sections.
  Si le bit le plus faible de cet élément est défini, les données de cette section seront transmises/reçues, voir *Redondance d'UC - Manuel de l'utilisateur*. (Ce bit correspond au point d'exclamation dans le navigateur de projet.)

Désactiver des sections

L'élément "disable" permet d'activer/de désactiver la section. Quand la variable multi-éléments n'est pas utilisée ou quand la valeur "0" a été attribuée à "disable", la section concernée est exécutée. Quand la valeur "1" a été attribuée à "disable", la section n'est pas exécutée. A l'aide de cette variable, l'exécution des sections peut être pilotée en fonction d'événements.

Note : Lors de l'animation d'une section désactivée, l'état INVALIDÉ apparaît dans la barre d'état.

⚠ ATTENTION

Risque d'états de traitement indésirables

Désactiver une section ne signifie pas que les sorties programmées sont désactivées au sein de la section. Si une sortie a déjà été définie dans un cycle précédent, cet état est conservé, même à l'issue de la désactivation de la section. L'état de ces sorties ne peut pas être modifié.

Le non-respect de ces instructions peut provoquer des blessures ou des dommages matériels.

Désactiver des sections d'interruption

Le bloc fonction ISECT_OFF permet de désactiver une certaine section d'interruption. Le bloc fonction ISECT_ON permet de l'activer à nouveau. La variable de commande SECT_CTRL fournit à cet effet le nom de la section.

Le bloc fonction I_LOCK permet de désactiver l'ensemble des sections d'interruption. Le bloc fonction I_UNLOCK permet de les activer toutes de nouveau.

Note : Une éventuelle interruption sur une section d'interruption inactive n'a aucune répercussion.

Invalidation INCONDITIONNELLE d'une section (possibilité 1)

Pour désactiver une section de façon inconditionnelle, effectuez les opérations suivantes :

Etape	Action
1	Ouvrez l'éditeur de données de référence (voir *Editeur des données de référence, p. 611*) à l'aide de **En ligne** → **Editeur de données de référence**.
2	Cliquez deux fois sur un numéro de ligne pour ouvrir la boîte de dialogue **Consulter variables**.
3	Dans la zone **Type de données**, sélectionnez dans un premier temps l'option **Structuré** puis, dans la zone de liste, **SECT_CTRL**. **Réaction :** Tous les noms de sections sont affichés.
4	Sélectionnez à présent le nom de la section à désactiver.
5	Par un clic sur **Composants...**, ouvrez la boîte de dialogue **Sélection des éléments de type ANY**.
6	Sélectionnez la ligne **disable : BOOL** et confirmez par **OK**.
7	Si cela n'a pas encore été effectué : Etablissez la connexion entre l'API et le PC et chargez votre programme dans l'API.
8	Modifiez l'entrée de la colonne **Valeur** et faites la passer à 1 (TRUE), en vue de désactiver la section, ou sur 0 (FALSE), afin de l'activer.
9	Si l'animation est inactive, activez-la à l'aide de **En ligne** → **Animer**. **Réaction :** En fonction de la valeur, la section est désactivée ou activée. **Note :** Désactiver une section ne signifie pas que les sorties programmées sont désactivées au sein de cette section. Si une sortie a déjà été définie dans un cycle précédent, cet état est conservé, même après désactivation de la section. L'état de ces sorties ne peut pas être modifié.

⚠ ATTENTION

Risque d'états de traitement indésirables

L'entrée de la colonne **Valeur** est conservée, même après que vous avez quitté l'éditeur de données de référence (même si les entrées n'ont pas été enregistrées). En d'autres termes, la section reste désactivée et elle doit être réactivée de façon explicite dans l'éditeur de données de référence (valeur = 0).

Le non-respect de ces instructions peut provoquer des blessures ou des dommages matériels.

Invalidation INCONDITIONNELLE d'une section (possibilité 2)

Pour désactiver une section de façon inconditionnelle, effectuez les opérations suivantes :

Etape	Action
1	Ouvrez le Navigateur de projet (voir *Navigateur de projet, p. 573*) à l'aide de **Projet** → **Navigateur de projet**.
2	Etablissez avec **En ligne** → **Connecter...** une connexion entre le PC et l'API.
3	Etablissez avec **En ligne** → **Chargement...** (en mode **DIFFERENTS**) ou **En ligne** → **Chargement des modifications...** (en mode **MODIFIE**) la cohérence entre le PC et l'API.
4	Dans le navigateur de projet, sélectionnez la section à désactiver.
5	Cliquez sur le bouton droit de la souris pour faire apparaître le menu contextuel des sections et activez le mode **Animer état de libération**.
6	Dans le menu contextuel (bouton droit de la souris) de la section sélectionnée, changez l'état de validation à l'aide de la commande **Changer l'état de libération**. **Note :** Le navigateur de projet vous permet uniquement de désactiver/de valider des sections n'étant pas déjà désactivées/validées par le biais d'une autre Section (voir *Désactivation CONDITIONNELLE d'une section, p. 59*) ou de l'éditeur de données de référence (voir *Invalidation INCONDITIONNELLE d'une section (possibilité 1), p. 57*). **Réaction :** Désactivation de la section. **Note :** Désactiver une section ne signifie pas que les sorties programmées sont désactivées au sein de cette section. Si une sortie a déjà été définie dans un cycle précédent, cet état est conservé, même après désactivation de la section. L'état de ces sorties ne peut pas être modifié.

Désactivation CONDITIONNELLE d'une section

Pour désactiver une section de façon conditionnelle (en fonction d'un programme), effectuez les opérations suivantes :

Etape	Action
1	Créez la logique selon laquelle la section doit être désactivée, dans une section FDB, par exemple. Lors de cette opération, tenez compte du fait que la logique doit obligatoirement livrer une "sortie" de type BOOL et que la section à désactiver l'est en cas d'un "1" logique. **Note :** Il ne faut pas désactiver la section contenant la logique destinée à l'invalidation/la validation d'autres sections.
2	Ouvrez la boîte de dialogue **Connecter FFB** par un double clic sur la "sortie" de votre logique.
3	Cliquez sur **Consulter...** pour ouvrir la boîte de dialogue **Consulter variables**.
4	Dans la zone **Type de données**, sélectionnez dans un premier temps l'option **Structuré** puis, dans la zone de liste, **SECT_CTRL**. **Réaction :** Tous les noms de sections sont affichés.
5	Sélectionnez à présent le nom de la section à désactiver par un double clic.
6	Sélectionnez la ligne **disable : BOOL** et confirmez par **OK**. **Réaction :** La variable multi-éléments (nom de section.disable) affectée à la section à désactiver est maintenant la "sortie" de votre logique.
7	Cliquez sur **Projet** → **Ordre d'exécution...** pour faire apparaître la boîte de dialogue **Ordre d'exécution des sections**.
8	Utilisez les boutons de commande pour garantir que la section contenant la logique d'invalidation sera exécutée avant la section à désactiver.
9	Si cela n'a pas encore été effectué : Etablissez la connexion entre l'API et le PC.
10	Chargez votre programme dans l'API. **Réaction :** Dans le cas d'un "1" logique, la section à désactiver n'est pas traitée. **Note :** Désactiver une section ne signifie pas que les sorties programmées sont désactivées au sein de cette section. Si une sortie a déjà été définie dans un cycle précédent, cet état est conservé, même après désactivation de la section. L'état de ces sorties ne peut pas être modifié.

Données de configuration

Description

La configuration de l'API est l'interface entre le programme et le matériel.

Les données de configuration se composent pour l'essentiel de la liste des composants et de l'indication des plages d'adresses pour le programme.

Les instructions chargeables sont destinées à l'échange de données avec le langage de programmation CEI et au chargement d'instructions LL984 supplémentaires.

Créer un projet

4

Introduction

Aperçu

Ce chapitre décrit la procédure générale à suivre lors de la première création d'un projet. Dans cette optique, nous avons opté pour une description la plus linéaire possible du déroulement des opérations, de façon à fournir une méthode claire aux nouveaux utilisateurs de Concept. Naturellement, les références croisées entre les commandes de menu sont possibles. Avec l'expérience, vous trouverez d'autres méthodes souvent plus courtes. Pour des informations plus détaillées, veuillez consulter les chapitres correspondants du manuel de l'utilisateur.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Vue d'ensemble	62
Etape 1 : démarrage de Concept	63
Etape 2 : Configuration automate	64
Etape 2.1 : Configuration requise	65
Etape 2.2 : Configuration optionnelle	68
Etape 3 : création du programme utilisateur	72
Etape 4 : Enregistrer	75
Etape 5 : Effectuez l' évaluation mémoire	76
Etape 6 : Chargement et test	77
Etape 7 : Optimisation et séparation	82
Etape 8 : Documentation	84

Vue d'ensemble

Créer un projet

La création d'un projet se divise en huit étapes principales:

Etape	Action
1	**Démarrage de Concept** (voir *Etape 1 : démarrage de Concept, p. 63*) Démarrez Concept et créez un nouveau projet.
2	**Configuration automate** (voir *Etape 2 : Configuration automate, p. 64*) Définissez la configuration matérielle.
3	**Création du programme utilisateur** (voir *Etape 3 : création du programme utilisateur, p. 72*) Créez de nouvelles sections et créez votre programme.
4	**Enregistrer** (voir *Etape 4 : Enregistrer, p. 75*) Enregistrez votre projet.
5	**Effectuez l' évaluation mémoire** (voir *Etape 5 : Effectuez l' évaluation mémoire, p. 76*) Vérifiez la part de mémoire occupée de l'API.
6	**Chargement et test du projet** (voir *Etape 6 : Chargement et test, p. 77*) Etablissez la connexion entre le PC et l'API. Chargez le projet dans l'API et démarrez celui-ci. Testez ensuite le programme à l'aide des fonctions de test en ligne. Eliminez à présent les erreurs éventuelles du programme. Chargez les sections modifiées dans l'API.
7	**Optimisation et séparation** (voir *Etape 7 : Optimisation et séparation, p. 82*) A présent, il est recommandé d'optimiser l'occupation de la mémoire de programme et de charger à nouveau le programme optimisé dans l'API. Une fois le chargement et l'éventuelle optimisation effectués, vous pouvez séparer le PC de l'API. A présent, le programme fonctionne hors ligne.
8	**Documentation** (voir *Etape 8 : Documentation, p. 84*) Créez une documentation complète de votre projet.

Notes

Note : Vous pouvez inverser l'ordre des étapes "Configuration automate" et " Création du programme utilisateur". En d'autres termes, il est également possible de modifier la configuration de l'API après la création du programme.

Note : Pour éviter des pertes de données, il est conseillé d'enregistrer régulièrement.

Etape 1 : démarrage de Concept

Démarrage de Concept

Pour démarrer Concept, effectuez les opérations suivantes :

Etape	Action
1	Pour démarrer Concept, cliquez deux fois sur l'icône de Concept.
2	Sélectionnez **Fichier** → **Nouveau projet**.
3	Définissez un nouveau chemin d'accès au projet ou reprenez le chemin d'accès proposé par défaut avec le nom de projet `sansnom.prj`. **Réaction :** Le nouveau projet s'ouvre. **Remarque :** Si vous reprenez le chemin d'accès par défaut avec le nom de projet `sansnom.prj`, vous pouvez toujours enregistrer*Etape 4 : Enregistrer, p. 75* ce projet ultérieurement sous un autre nom. Un projet déjà enregistré est appelé à l'aide du menu **Ouvrir projet...** ou via l'icône du projet.

Remarque

Note : Observez, pour toutes les étapes ultérieures, les réglages dans le sous-menu de **Options** → **Environnement**!

Suite

Continuez ensuite avec l'étape 2 : configuration de l'automate (voir *Etape 2 : Configuration automate, p. 64*).

Etape 2 : Configuration automate

Qu'est ce qu'il faut configurer ?

Les commandes **Projet** → **Configuration automate** vous permettent de configurer toute la configuration matérielle de votre projet.

Configuration requise

Note : Le type d'API doit d'abord être défini ! Toutes les autres configurations peuvent ensuite être exécutées indépendamment de l'ordre de traitement.

Les configurations suivantes sont nécessaires pour la planification :

- *Définition du type d'API (Configuration minimale), p. 65*
- *Définir la partition mémoire, p. 66*
- *Installer chargeables, p. 66*
- *Définir les affectations d'E/S, p. 67*

Configuration optionnelle

Les configurations suivantes doivent être utilisées en fonction du projet :

- *Définir station décentralisée d'E/S, p. 68*
- *Définir la communication Modbus, p. 68*
- *Définition de la communication de diffusion des E/S, p. 69*
- *Définir la protection d'accès aux données, p. 70*
- *Divers réglages API, p. 70*
- *Messages ASCII (uniquement pour 984 LL), p. 71*

Etape 2.1 : Configuration requise

Condition

Le type d'API doit d'abord être défini ! Toutes les autres configurations peuvent ensuite être exécutées indépendamment de l'ordre d'exécution.

Définition du type d'API (Configuration minimale)

Pour définir le type d'API (Configuration minimale), effectuez les opérations suivantes :

Etape	Action
1	Sélectionnez **Projet** → **Configuration automate**. **Réaction :** La fenêtre **Configuration automate** s'ouvre, elle contient d'autres commandes de menu pour la configuration du matériel.
2	Dans la zone de liste, sélectionnez **Sélection automate**. **Réaction :** La boîte de dialogue **Sélection automate** s'ouvre.
3	Dans la zone de liste **Gamme d'automate**, sélectionnez votre type d'API.
4	Dans la zone de liste **UC/exécutable**, sélectionnez votre UC.
5	Dans la zone de liste **Moteur d'exécution**, sélectionnez l'état **Validée** **Réaction :**La programmation de sections dans les langages de programmation CEI (FDB, LD, SFC, IL et ST) est possible. **Note :** Si l'état **Moteur d'exécution Non disponible**, **Invalider** ou **984 seul** est affiché, l'UC sélectionnée n'accepte pas de langage de programmation CEI . Si l'état **CEI seul** est affiché dans la zone de liste, l'UC sélectionnée accepte uniquement les langages CEI et il n'est donc pas nécessaire de libérer ceux-ci de façon explicite.
6	Vous pouvez, pour des programmes et des essais simples, quitter la configuration et passer à l'*Etape 3 : création du programme utilisateur, p. 72* ou *Etape 4 : Enregistrer, p. 75* continuer.

Définir la partition mémoire

Pour définir la partition mémoire, effectuez les opérations suivantes :

Etape	Action
1	Sélectionnez **Projet** → **Configuration automate**. **Réaction :** La fenêtre **Configuration automate** s'ouvre, elle contient d'autres commandes de menu pour la configuration du matériel.
2	Dans la zone de liste, sélectionnez **Partition mémoire de l'API**. **Réaction :** La boîte de dialogue **Partition mémoire de l'API** s'ouvre.
3	Dans les zones **Bits** et **Mots** sélectionnez le nombre probable de bits mémento d'E/S et de mots d'E/S dont le programme utilisateur aura besoin. **Note :** La plage maximale d'adresses ne devant pas être dépassée est affichée au niveau du côté droit de la boîte de dialogue.

Installer chargeables

Pour installer les chargeables, effectuez les opérations suivantes :

Etape	Action
1	Sélectionnez **Projet** → **Configuration automate**. **Réaction :** La fenêtre **Configuration automate** s'ouvre, elle contient d'autres commandes de menu pour la configuration du matériel.
2	Dans la zone de liste, sélectionnez **Instructions chargeables**. **Réaction :** Le dialogue **Chargeables** s'ouvre.
3	Sélectionnez dans la zone de liste **Disponible :** l'instruction chargeable. **Note :** L'affection des chargeables est indiquée à la section *Instructions chargeables, p. 106*.
4	Sélectionnez le bouton de commande **Installer =>**. **Réaction :** Le chargeable sélectionné est placé dans le champ **Installé :**
5	Répétez les étapes 3 et 4 , jusqu'à ce que tous les chargeables soient installés.

Définir les affectations d'E/S

Pour définir les affectations d'E/S, suivez les étapes suivantes :

Etape	Action
1	Sélectionnez **Projet** → **Configuration automate**. **Réaction :** La fenêtre **Configuration automate** s'ouvre, elle contient d'autres commandes de menu pour la configuration du matériel.
2	Dans la zone de liste, sélectionnez **Affectations d'E/S**. **Réaction :** La boîte de dialogue **Affectations d'E/S** s'ouvre.
3	Dans la colonne **Maintient** (temps de contrôle), entrez la durée pendant laquelle un échange de communication doit se produire. Si cette durée est dépassée, une erreur est signalisée.
4	Sélectionnez le bouton de commande **Editer...**. **Réaction :** La boîte de dialogue pour entrer les groupes de travail s'ouvre.
5	Sélectionnez dans la colonne **Module** le bouton de commande **...**. **Réaction :** Le dialogue **Sélection de modules d'E/S** s'ouvre.
6	Sélectionnez dans la colonne **Modules** le module. **Réaction :** Le module est affiché au niveau de l'emplacement actuel.
7	Sélectionnez les colonnes **Déb.entr** ou **Déb.sort** et entrez pour le module la première adresse de la plage de référence de sortie ou d'entrée occupée.
8	Sélectionnez le module et ensuite le bouton de commande **Paramètres**. **Réaction :** Si le module dispose d'un dialogue de paramétrage, vous pouvez définir ici les paramètres (par ex. comportement de coupure, format des données, plage de mesure).

Suite

Continuez ensuite avec l'étape 3 : Création du programme utilisateur
(voir *Etape 3 : création du programme utilisateur, p. 72*).

Etape 2.2 : Configuration optionnelle

Remarques générales

Les configurations suivantes ne doivent pas impérativement être effectuées, mais offrent plutôt des fonctions étendues.

Définir station décentralisée d'E/S

Pour définir les E/S décentralisées, effectuez les opérations suivantes (pour la Configuration minimale, cette procédure est facultative) :

Etape	Action
1	Sélectionnez **Projet** → **Configuration automate**. **Réaction :** La fenêtre **Configuration automate** s'ouvre, elle contient d'autres commandes de menu pour la configuration du matériel.
2	Dans la zone de liste, sélectionnez **Affectations d'E/S**. **Réaction :** La boîte de dialogue **Affectations d'E/S** s'ouvre.
3	Sélectionnez le bouton de commande **E/S déportées...**. **Réaction :** La boîte de dialogue **E/S déportées** s'ouvre.
4	Indiquez les emplacements pour les modules RIO ou NOM. **Réaction :** Vous retournez au dialogue **Affectations d'E/S**.
5	Dans la zone de liste **Aller à**, sélectionnez l'E/S décentralisée.
6	Sélectionnez dans le tableau une ligne vide (dernière ligne), et sélectionnez le bouton de commande **Insérer**. **Réaction :** Une station d'E/S supplémentaire est inscrite dans la colonne **Type** .
7	Sélectionnez dans la colonne **St. E/S** et entrez le numéro de la station. **Note :** Il est possible de configurer au maximum autant de stations d'E/S décentralisées qu'il y a de segments inscrits dans l'ordonnanceur de segments.
8	Dans la zone de liste **Aller à**, sélectionnez l'E/S décentralisée pour la deuxième station d'E/S.
9	Exécutez ensuite les étapes 3 à 8 de la procédure *Définir les affectations d'E/S, p. 67* .

Définir la communication Modbus

Pour définir la communication Modbus (esclave Quantum, terminal, imprimante, ...), exécutez la commande suivante :

Etape	Action
1	Sélectionnez **Projet** → **Configuration automate**. **Réaction :** La fenêtre **Configuration automate** s'ouvre, elle contient d'autres commandes de menu pour la configuration du matériel.
2	Dans la zone de liste, sélectionnez la commande **Configuration port Modbus**. **Réaction :** Le dialogue **Configuration port Modbus** s'ouvre.
3	Procédez aux réglages correspondants.

Définition de la communication de diffusion des E/S

Si une connexion Modbus Plus existe, la fonctionnalité de diffusion des E/S vous permet de transmettre des données de mémoire d'état au niveau du réseau local, de façon globale, ou directement entre plusieurs abonnés de bus. Pour définir la communication de diffusion des E/S, effectuez les opérations suivantes :

Etape	Action
1	Sélectionnez **Projet** → **Configuration automate**. **Réaction :** La fenêtre **Configuration automate** s'ouvre, elle contient d'autres commandes de menu pour la configuration du matériel.
2	Dans la zone de liste, sélectionnez **Extensions de config.** → **Sélectionner extensions**. **Réaction :** La boîte de dialogue **Sélectionner extensions** s'ouvre.
3	Activez la case à cocher **Diffusion des E/S**. **Réaction :** Vous revenez sur la fenêtre **Configuration automate** , et la commande de menu **Diffusion des E/S** est à présent disponible.
4	Sélectionnez **Extensions de config.** → **Diffusion des E/S**. **Réaction :** Le dialogue **Diffusion des E/S** s'ouvre.
5	Dans la zone **Aller à** , sélectionnez l'abonné de bus local et entrez l'emplacement.
6	Dans la zone **Données globales** , sélectionnez les boutons de commande **Réception...** et **Emission...**, pour définir les adresses source et cibles des données de transmission ou l'adresse de l'autre abonné de bus.
7	Dans la zone **Données directes** , sélectionnez les boutons de commande **Réception...** et **Emission...**, pour définir les adresses source et cibles des données de transmission ou l'adresse de l'autre abonné de bus.

Définir la protection d'accès aux données

Vous pouvez protéger des plages d'adresses de bits de sortie/de capteur et de mots de sortie/de capteur contre l'écrasement par signaux externes. Pour définir la protection d'accès aux données de données, effectuez les opérations suivantes :

Etape	Action
1	Sélectionnez **Projet** → **Configuration automate**. **Réaction :** La fenêtre **Configuration automate** s'ouvre, elle contient d'autres commandes de menu pour la configuration du matériel.
2	Dans la zone de liste, sélectionnez **Extensions de config.** → **Sélectionner extensions**. **Réaction :** Le dialogue **Sélectionner extensions** s'ouvre.
3	Activez la case à cocher **Protection des données**. **Réaction :** Vous revenez sur la fenêtre **Configuration automate** , et la commande de menu **Protection en écriture de la mémoire des données** est à présent disponible.
4	Sélectionnez **Extensions de config.** → **Protection en écriture de la mémoire de données**. **Réaction :** La boîte de dialogue **Protection en écriture de la mémoire de données** s'ouvre.
5	Sélectionnez les bits de sortie/mémento et les mots de sortie/mémento pour la zone devant recevoir une protection d'écriture.

Divers réglages API

Diverses données internes d'API peuvent être utilisées, un contrôle de l'horloge du programme utilisateur (chien de garde, timeout) peut être défini, la fenêtre de temps de communication (tranche de temps d'E/S) peut être paramétrée, et l'attribution multiple de sorties peut être autorisée. Pour définir ces réglages API, effectuez les opérations suivantes :

Etape	Action
1	Sélectionnez **Projet** → **Configuration automate**. **Réaction :** La fenêtre **Configuration automate** s'ouvre, elle contient d'autres commandes de menu pour la configuration du matériel.
2	Dans la zone de liste, sélectionnez **Divers**. **Réaction :** Le dialogue **Divers** s'ouvre.
3	Activez la case à cocher **Sortie pile**, **Registre base de temps** et **Date/Heure**, et entrez une adresse dans les zones de texte correspondantes.
4	Activez la case à cocher **Autoriser bobines doubles (LL984 seulement)**, et entrez dans la zone de texte l'adresse à partir de laquelle ceci est permis.
5	Entrez dans la zone de texte **Chien de garde (ms*10) :** une valeur numérique entre 2 et 255 (ms). Vous pouvez ainsi spécifier une surveillance par impulsions du programme utilisateur. **Réaction :** Dès que les impulsions de comptage sont absentes pour un temps plus long que le temps défini, un message d'erreur apparaît.
6	Entrez dans la zone de texte **Tranche de temps des modifications en ligne (ms) :** une valeur numérique entre 3 et 100 (ms). Vous pouvez ainsi définir le temps pour l'exécution des fonctions d'édition multicycles (Insérer, Effacer, Rechercher etc.).

Messages ASCII (uniquement pour 984 LL)

Pour définir les messages ASCII (uniquement pour 984 LL), effectuez les opérations suivantes :

Etape	Action
1	Sélectionnez **Projet** → **Configuration automate**. **Réaction :** La fenêtre **Configuration automate** s'ouvre, elle contient d'autres commandes de menu pour la configuration du matériel.
2	Sélectionnez dans la zone de liste **ASCII** → **Installation ASCII**. **Réaction :** La boîte de dialogue **Installation ASCII** s'ouvre.
3	Entrez dans les zones de texte le nombre total de messages, la taille des messages et le nombre de ports ASCII (de la périphérie d'E/S). **Réaction :** Dans la fenêtre **Configuration automate** → **ASCII** la commande de menu **Configuration port ASCII** est disponible.
4	Dans la zone de liste, sélectionnez la commande **ASCII** → **Configuration port ASCII**. **Réaction :** Le dialogue **Configuration port ASCII** s'ouvre.
5	Procédez aux réglages correspondants. **Note :** Les messages ASCII peuvent être générées sous **Projet** → **Messages ASCII...**

Suite

Continuez ensuite avec l'étape 3 : Création du programme utilisateur
(voir *Etape 3 : création du programme utilisateur, p. 72*).

Etape 3 : création du programme utilisateur

Généralités

Le programme utilisateur est créé par sections. Chaque section peut être programmée dans l'un des langages de programmation disponibles. Chaque section a un nom unique au sein du projet. Pendant la programmation, il est possible de créer des sections à tout moment.

Vue d'ensemble

La création du programme utilisateur comporte neuf étapes principales :

Etape	Action
1	Création d'une nouvelle section (voir *Création d'une nouvelle section, p. 72*)
2	Déclaration des variables (voir *Déclaration des variables, p. 73*)
3	Programmation d'une section (voir *Programmation d'une section, p. 73*)
4	Analyse du programme/d'une section (voir *Analyse du programme/d'une section, p. 74*)
5	Définition de l'ordre d'exécution des sections (voir *Définition de l'ordre d'exécution des sections, p. 74*)

Création d'une nouvelle section

Pour créer une nouvelle section, effectuez les opérations suivantes :

Etape	Action
1	Appelez dans le menu principal **Fichier** la commande **Nouvelle section...** . **Réaction :** La boîte de dialogue **Nouvelle section** s'ouvre.
2	A l'aide de la souris, sélectionnez le langage de programmation souhaité pour cette section.
3	Dans la zone de texte **Nom de section**, entrez le nom unique de cette section.
4	Utilisez la même méthode pour créer toutes les sections nécessaires.

Déclaration des variables

Un programme se compose de fonctions et de blocs fonctions (FFB) ou d'instructions avec indication de variables (signaux, par exemple), d'adresses directes ou de littéraux. Alors que les adresses directes et les littéraux peuvent être immédiatement utilisés, les variable, elles, doivent obligatoirement être déclarées avant de pouvoir être utilisées lors de la programmation. Pour déclarer les variables, effectuez les opérations suivantes :

Etape	Action
1	Appelez dans le menu principal **Projet** la commande **Déclaration de variables...** . **Réaction :** La boîte de dialogue **Editeur de variable** s'ouvre.
2	Entrez le nom des variables, leur type de données et le cas échéant leur adresse de référence, leur valeur initiale, ainsi qu'un éventuel commentaire.
3	Confirmez les entrées à l'aide de **OK**. **Note :** Une édition ultérieure est également possible en cliquant deux fois à partir d'une connexion FFB ou d'un contact, etc. -> **Déclaration var...** possible. L'éditeur de variables est alors démarré.

Programmation d'une section

Pour programmer une section, effectuez les opérations suivantes :

Etape	Action
1	Ouvrez la section à programmer à l'aide de **Fichier** → **Ouvrir section**.
2	Créez votre programme en respectant les règles des différents langages de programmation : ● FDB (langage de bloc fonction) (voir *Langage des blocs fonctions FDB, p. 211*) ● LD (langage à contacts (CEI)) (voir *Plan de contact LD, p. 237*) ● SFC (automate d'enchaînement) (voir *Langage de commande séquentielle SFC, p. 273*) ● IL (liste d'instructions) (voir *Liste d'instructions IL, p. 325*) ● ST (littéral structuré) (voir *Texte structuré ST, p. 399*) ● LL984 (langage à contacts (Modsoft)) (voir *Ladder Logic 984, p. 463*)

Analyse du programme/ d'une section

Recherchez et éliminez les erreurs de syntaxe contenues dans une section ou dans le programme entier ! Pour analyser une section/un programme, effectuez les opérations suivantes :

Etape	Action
1	Dans le menu principal **Projet**, appelez la commande **Analyser section** ou **Analyser programme**.
2	Eliminez la cause des erreurs affichées ou signalées. **Note :** Seule l'absence d'erreurs détectées lors de la vérification permet de charger la section ou le programme dans l'API. (L'élimination de la cause d'avertissements n'est pas nécessaire. Une vérification des messages d'erreur est conseillée.)

Définition de l'ordre d'exécution des sections

Dans un premier lieu, les sections sont stockées dans leur ordre de création, puis exécutées après le démarrage du programme. En général, il faut adapter cet ordre aux tâches étant spécifiques au projet. Pour définir l'ordre d'exécution des sections, effectuez les opérations suivantes :

Etape	Action
1	Pour définir l'ordre d'exécution des section, on distingue deux possibilités : ● Appelez dans le menu principal **Projet** la commande **Ordre d'exécution...** et placez toutes les sections dans l'ordre souhaité, à l'aide des boutons **Première**, **Dernière**, **Suivante**, **Précédente**. ● Dans le menu principal **Projet**, appelez la commande **Navigateur de projet** et déplacez les sections pour les ordonnez comme vous le souhaitez dans le *Navigateur de projet, p. 573*.

Suite

Continuez ensuite avec l'étape 4 : Enregistrer (voir *Etape 4 : Enregistrer, p. 75*).

Etape 4 : Enregistrer

Remarques générales

Remarques générales sur l'enregistrement :

- Si vous quittez un projet sans l'avoir enregistré auparavant, vous êtes automatiquement invité à indiquer si vous voulez effectuer un enregistrement ou non. Si vous répondez Oui à la question, le déroulement est le même que celui décrit ci-dessous.
- Pour éviter des pertes de données, il est conseillé d'effectuer régulièrement des enregistrements pendant les sessions de configuration ou de programmation de longue durée.

Premier enregistrement d'un projet

Pour enregistrer un projet pour la première fois, effectuez les opérations suivantes :

Etape	Action
1	Appelez dans le principal **Fichier** la commande **Enregistrer projet sous...** .
2	Entrez le nom de projet nom.prj dans la zone de texte **Nom de fichier**.
3	Sélectionnez le lecteur et le répertoire souhaités dans la zone de liste **Répertoires**. Une autre possibilité consiste à entrer le chemin d'accès intégral dans la zone de texte **Nom de fichier**, par exemple **`c : \product1\reactor3.prj.`** `(28 caractères au maximum + .prj)`. Les répertoires non existants sont automatiquement créés. **Note :** Selon la définition CEI 1131, un projet contient tous les programmes, toutes les données, etc. associés à un API. Si plusieurs projets (plusieurs API) sont associés à une installation, vous devez enregistrer tous les projets dans un répertoire commun portant le nom de l'installation.
4	Cliquez sur le bouton de commande **OK**. **Réaction :** Votre projet est à présent enregistré dans le répertoire indiqué, sous le nom que vous avez indiqué.

Enregistrements suivants

Pour continuer à enregistrer, effectuez les opérations suivantes :

Etape	Action
1	Il vous suffit d'utiliser la commande **Enregistrer** du menu principal **Fichier**.

Suite

Continuez ensuite avec l'étape 5 : Exécution de l'evaluation mémoire (voir *Etape 5 : Effectuez l' évaluation mémoire, p. 76*).

Etape 5 : Effectuez l' évaluation mémoire

Vérifiez la part de mémoire occupée.

Avant le téléchargement du programme sur l'API, il faudrait effectuer une prédiction mémoire hors ligne de l' automate configuré. La table affichée dans le dialogue **Projet** → **Prédiction mémoire** indique l' utilisation des différentes zones de mémoires. Vous connaissez ainsi à l'avance la part de mémoire occupée.

Note : L'évaluation mémoire n'est pas très précise dans certains cas. Il peut y avoir une différence de plus de 20 % entre la mémoire utilisée dans l'automate et l'évaluation mémoire obtenue sous Concept. L'évaluation mémoire indique toujours un espace mémoire plus important que celui réellement disponible dans l'automate.
Cela est dû à la difficulté du calcul de la mémoire dynamique dans les DFB et sections. Les sections ST sont en particulier la cause d'une grande différence entre l'évaluation et l'automate. Afin de s'assurer de disposer de suffisamment de mémoire dans l'automate, il faut charger le projet dans un automate afin des fins de test. Le simulateur ne peut pas être utilisé car beaucoup de projets ont suffisamment d'espace dans le simulateur mais pas suffisamment dans l'automate.

Suite

Continuez ensuite avec l’étape 6 : Chargement et test du projet
(voir *Etape 6 : Chargement et test, p. 77*).

Etape 6 : Chargement et test

Remarques générales

Il est possible de charger et de tester le programme seulement si une des conditions suivante est remplie :

- soit le simulateur Concept-SIM à 16 bits est activé, soit
- le simulateur 16 bits SIM Concept est désactivé et un API est connecté via un câble Modbus Plus, Modbus ou TCP/IP.
- Le simulateur PLCSIM32 Concept est activé.

Note : C'est seulement dans le cas de Programmes Utilisateur CEI qu'il est possible d'effectuer les tests avec les simulateurs SIM Concept (voir *Simuler un API (Simulateur 16 bits), p. 775*) et PLCSIM32 Concept (voir *Simuler un API (Simulateur 32 bits), p. 777*).

Vue d'ensemble

Le chargement et les essais de programmes comprennent neuf étapes principales :

Etape	Action
1	Chargement du fichier EXEC dans l'API (voir *Instructions d'installation de Concept*)
2	Connexion de l'ordinateur et de l'API (voir *Connexion de l'ordinateur et de l'API, p. 78*)
3	Chargement et démarrage du programme (voir *Chargement et démarrage du programme, p. 78*)
4	Activation de l'animation (voir *Activation de l'animation, p. 79*)
5	Modification des valeurs de libellés (voir *Modification des valeurs de libellés, p. 79*)
6	Modification des valeurs de variables. (voir *Modification des valeurs de variables., p. 80*)
7	Localisation des erreurs survenues (voir *Localisation des erreurs survenues, p. 80*)
8	Chargement des modifications (voir *Chargement des modifications, p. 81*)
9	Démarrage et arrêt de l'API (voir *Démarrage et arrêt de l'API, p. 81*)

Connexion de l'ordinateur et de l'API

Pour connecter le PC et l'API, effectuez les opérations suivantes :

Etape	Action
1	Appelez dans le menu principal **En ligne** la commande **Connecter...** . **Réaction :** La boîte de dialogue **Connecter à l'API** s'ouvre.
2	Déterminez le type de protocole (Modbus, Modbus Plus, TCP/IP ou Simulateur) ainsi que l'abonné API (si vous travaillez en réseau) avec lequel vous souhaitez communiquer.
3	Sélectionnez sous **Niveau d'accès** l'option **Modifier la configuration**.
4	Confirmez vos entrées à l'aide de **OK**.

Chargement et démarrage du programme

Pour charger et démarrer le programme, effectuez les opérations suivantes :

Etape	Action
1	Appelez dans le menu principal **En ligne** la commande **Chargement...** . **Réaction :** La boîte de dialogue **Charger dans l'automate**s'ouvre.
2	Lorsqu'il s'agit du premier chargement, cliquez sur le bouton de commande **Tous**.
3	Cliquez sur le bouton de commande **Charger**. **Réaction :** Diverses boîtes de dialogue sont affichées.
4	Répondez à la question s'il faut `Arrêter le programme dans l'API Oui/Non` avec **Oui**. **Note :** Cette question apparaît seulement si un programme est déjà actif sur votre API.
5	Répondez à la question s'il faut `Lancer le programme dans l'API Oui/Non` avec **Oui**, si aucune erreur ne s'est produite. Si des avertissements ou des erreurs ont été signalisés, ceux-ci sont alors répertoriés dans la fenêtre **Messages**. Corrigez les avertissements/erreurs à l'endroit indiqué.

Activation de l'animation

L'animation (affichage de l'état en ligne) vous permet d'afficher l'état de variables, d'étapes, de transitions, etc. dans la fenêtre d'éditeur de sections individuelles. Pour activer l'animation, effectuez les opérations suivantes :

Si ...	Alors ...
Si vous ne voulez afficher que des valeurs binaires.	Pour afficher exclusivement des valeurs binaires, appelez la commande **Animer valeurs binaire** du menu principal **En ligne**. **Réaction :** Les valences de toutes les valeurs binaires (variables, adresses directes, libellés) sont affichés en couleur (signal 0 = rouge, signal 1 = vert).
Si vous voulez afficher les valeurs de toutes les variables.	Pour afficher les valeurs de toutes les variables, appelez tout d'abord la commande **Sélectionner tout** (sélectionner tous les éléments de la section actuelle) du menu principal **Edition**. Appelez ensuite la commande **Animer sélection** du menu principal **En ligne**. **Réaction :** Les valences de toutes les valeurs (variables, adresses directes, libellés) sont affichées en couleur (rouge = signal 0, vert = signal 1, jaune = affichage immédiat de la valeur pour les variables, et affichage des valeurs en cliquant deux fois sur la variable pour les variables multi-éléments).
Si vous voulez insérer des champs de contrôle dans les langages texte (IL et ST).	La commande **Sélection dans champ d'inspection** vous permet, dans les langages texte IL et ST, d'insérer des champs de contrôledans la section. **Réaction :** La valeur actuelle des variables affectées est affichée dans ces champs de surveillance. Dans le cas de variables multi-éléments, seule la valeur du premier élément est affichée. Pour modifier cela, cliquez deux fois sur le champs de contrôle. Cette opération vous permet d'accéder à la boîte de dialogue **Paramétrage des champs de contrôle** et aux options de celui-ci.

Modification des valeurs de libellés

Pour modifier des libellés, effectuez les opérations suivantes :

Etape	Action
1	Activez l'animation, comme décrit dans *Activation de l'animation, p. 79*.
2	Cliquez deux fois sur le libellé à modifier.
3	Entrez une nouvelle valeur et confirmez avec **OK**. **Réaction :** La nouvelle valeur est transférée dans l'API lors du cycle suivant.

Modification des valeurs de variables.

L'Editeur de données de référence (voir *Editeur des données de référence, p. 611*) vous permet d'afficher et de paramétrer des valeurs de variable (état, commander, forcer). Pour modifier des variables, effectuez les opérations suivantes :

Etape	Action
1	Dans le menu principal **En ligne**, appelez la commande **Editeur de données de référence**.
2	Dans la boîte de dialogue **Tableaux Editeur de données**, entrez la variable à afficher.
3	Pour définir des valeurs, sélectionnez la case à cocher **Invalider** et entrez la valeur souhaitée.
4	Il est possible d'enregistrer le tableau ED sous un nom unique. Pour cela, sélectionnez dans le menu principal **Editeur de données** la commande **Enregistrer tableau sous...** . **Note :** Il est possible d'appeler plusieurs Tableaux Editeur de donnéesiteur de données en même temps. Pour cela, appelez la commande **Ouvrir tableau...** du menu principal **Editeur de données**.

Localisation des erreurs survenues

Si des erreurs surviennent lors du traitement du programme sur l'API, elles sont signalisées dans la fenêtre **Messages** et elles sont consignées sous forme de journal dans une liste des événements. Pour localiser des erreurs survenues, effectuez les opérations suivantes :

Etape	Action
1	Appelez dans le menu principal **En ligne** la commande **Affichage des évènements...** . **Réaction :** Une fenêtre s'ouvre. Dans celle-ci, toutes les erreurs survenues sont répertoriées et décrites.
2	Sélectionnez une ligne d'erreur, puis cliquez sur le bouton de commande **Atteindre l'erreur.**. **Réaction :** Vous atteignez directement la section dans laquelle l'erreur est survenue. L'objet incorrect et sélectionné.
3	Corrigez le programme.
4	Si votre programme se trouve à présent dans l'état **INEGAUX**, exécutez à nouveau les étapes de Chargement et démarrage du programme (voir *Chargement et démarrage du programme, p. 78*). Si votre programme se trouve à présent dans l'état **MODIFIE**, exécutez à nouveau les étapes de Chargement des modifications (voir *Chargement des modifications, p. 81*).

Chargement des modifications

Si l'état du projet est **MODIFIE** après l'exécution des modifications, ces modifications peuvent être chargées en ligne dans l'API, sans qu'il soit nécessaire d'arrêter le programme en cours d'exécution. Pour charger des modifications, effectuez les opérations suivantes :

Etape	Action
1	Appelez dans le menu principal **En ligne** la commande **Chargement des modifications...** .
2	Cliquez sur **OK**. **Réaction :** Les modifications sont chargées dans l'API.

Démarrage et arrêt de l'API

Pour démarrer et arrêter l'API, effectuez les opérations suivantes :

Etape	Action
1	Si le même projet est lancé sur l'ordinateur et l'API (**EGAUX**), l'API peut être redémarré ou arrêté avec **En ligne** → **Commande en ligne...**

Suite

Continuez ensuite avec l'étape 7 : Optimisation et séparation
(voir *Etape 7 : Optimisation et séparation, p. 82*) :

Etape 7 : Optimisation et séparation

Optimisation de projets

Après avoir mis en marche ou exécuté plusieurs fois la commande **Chargement des modifications...**, il est recommandé d'exécuter une optimisation, de façon à refermer les espaces apparus dans la gestion mémoire des données de programmation. Le projet du PC et l'API étant **INEGAUX** après l'optimisation, chargez l'application dans l'API grâce à la commande **Chargement...** (**Attention:** le programme doit être arrêté puis redémarré !). Pour optimiser les projets, effectuez les opérations suivantes :

Etape	Action
1	Enregistrez le projet à l'aide de **Fichier** → **Enregistrer projet**.
2	Dans le menu principal **Fichier**, appelez la commande **Fermer projet** et suivez les instructions des boîtes de dialogue successives.
3	Appelez dans le menu principal **Fichier** la commande **Optimiser projet...** et sélectionnez le projet à optimiser . Suivez les instructions des boîtes de dialogue successives.
4	Utilisez la commande **Statistiques mémoire...** du menu principal **En ligne** pour contrôler la taille de la mémoire de données de programme.
5	Adaptez éventuellement les tailles, à l'aide de **Configuration automate**.
6	Enregistrez le projet à l'aide de **Fichier** → **Enregistrer projet**.
7	Chargez à nouveau le programme optimisé dans l'API. Pour cela, utilisez **En ligne** → **Chargement...** . Pour cela, le programme en cours doit être arrêté.
8	Démarrez le programme nouvellement chargé à l'aide de **En ligne** → **Commande en ligne**.

Séparation de l'ordinateur et de l'API

Une fois le programme testé avec succès dans l'API (avec procédé connecté), le PC peut être séparé de l'API. Pour séparer le PC et l'API, effectuez les opérations suivantes :

Etape	Action
1	Tenez compte de l'état de programme affiché au bas de page ! Pour des raisons de cohérence, on doit impérativement avoir **EGAUX**. ● Si l'état est **MODIFIE**, vous devez d'abord charger les modifications *Chargement des modifications, p. 81*. ● Si l'état est **INEGAUX**, vous devez charger à nouveau le programme dans l'API *Chargement et démarrage du programme, p. 78*.
2	Appelez dans le menu principal **En ligne** la commande **Déconnecter** . Suivez les indications fournies dans les boîtes de dialogue affichées.
3	Après la séparation, le projet peut être fermé. Dans le menu principal **Fichier**, appelez la commande **Fermer projet**. Suivez les indications fournies dans les boîtes de dialogue éventuellement affichées.

Suite

Continuez ensuite avec l'étape 8 : Documentation (voir *Etape 7 : Optimisation et séparation, p. 82*).

Etape 8 : Documentation

Remarques générales

Il est conseillé de documenter chaque projet de façon complète. Les modifications et le ajouts doivent aussi être documentés (documentation partielle).

La documentation comprend, entre autres :

- des commentaires concernant le projet (**Projet** → **Caractéristiques**),
- des commentaires concernant les sections individuelles (**Fichier** → **Caractéristiques section**),
- des commentaires concernant les variables,
- des commentaires concernant les fonctions, les blocs fonctions et les DFB utilisés (bouton de commande **Commentaire** dans les dialogues de propriétés des différents blocs),
- des commentaires concernant les étapes et les transitions (bouton de commande **Commentaire** dans les dialogues de propriétés des différents éléments),
- des commentaires sous forme d'éléments de texte pouvant être librement placés dans les langages de programmation graphiques (**Objet** → **Texte**),
- des commentaires concernant les différentes lignes de commande dans les langages de programmation textuels,
- des commentaires concernant les types de données personnalisés,
- des commentaires concernant les blocs fonctions dérivés (DFB).

Impression de la documentation

Pour imprimer la documentation, effectuez les opérations suivantes :

Etape	Action
1	Appelez dans le menu principal **Fichier** la commande **Imprimer...** .
2	Dans la boîte de dialogue **Contenu des documents**, au niveau de **Présentation de la page**, effectuez une sélection pour indiquer si une en-tête et un pied de page uniformes doivent être imprimés pour chaque page. Indiquez également si une page de titre doit être imprimée. L'aspect des de l'en-tête, du pied de page, ainsi que de la page de titre est stocké dans les fichiers ASCII à sélectionner.
3	Dans la zone **Sélectionner documents** de la boîte de dialogue **Contenu des documents**, sélectionnez ce qui doit être imprimé.
4	Si vous avez sélectionné **Liste de variables**, vous devez appeler **Options**, en vue de sélectionner les variables à imprimer.
5	Si vous avez sélectionné **Sections**, ● vous devez appeler **Sélectionner...** et définir les sections à imprimer, et ● vous devez également appeler **Options**. Dans la zone **Facteur d'agrandissement des graphiques**, définissez entre autres la taille correcte de la logique à imprimer.
6	Cliquez sur le bouton de commande **OK**. **Réaction :** Toutes les entrées sont enregistrées.
7	Assurez-vous que la répartition des sections sur la page correspond à ce que vous souhaitez. Pour cela, appelez dans le menu principal **Affichage** la commande **Vue d'ensemble**, puis la commande **Saut de page**.
8	Modifiez l'agencement (des FFB, par exemple), de façon à ce qu'il y ait le moins de transitions possibles sur les pages adjacentes.
9	Apple Appelez à nouveau dans le menu principal **Fichier** la commande **Imprimer...** et confirmez avec le bouton de commande **Imprimer**. L'impression est effectuée selon les réglages définis et la boîte de dialogue est fermé.

Configuration API

5

Introduction

Aperçu

Cette section décrit les différentes séquences de configuration matérielle.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
5.1	Remarques générales sur la configuration matérielle	89
5.2	Configuration en mode local et connecté	93
5.3	Configuration inconditionnelle	97
5.4	Configuration optionnelle	115
5.5	Backplane Expander Config	127
5.6	Configuration de différents systèmes de réseau	131
5.7	Paramètres de sécurité Quantum dans le configurateur	145

5.1 Remarques générales sur la configuration matérielle

Introduction

Aperçu

Cette section vous donne un aperçu général de la configuration matérielle.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Remarques générales	90
Procédez ainsi lors de la configuration	91

Remarques générales

Introduction

Les incidences de la configuration Système sont très ramifiées, étant donné que celle-ci conditionne l'ensemble du fonctionnement de la gestion. La configuration a pour rôle de définir toutes les informations spécifiques de gestion, ainsi que celles d'ordre général, d'assigner les mémoires et de déterminer les zones d'entrée/sortie. Lors de la configuration initiale, l'utilisateur doit entrer un certain nombre de données fondamentales concernant le domaine API, tel que Type API et Mémoire. Seules les configurations en vigueur sont admises.

Une configuration se réfère toujours à un Projet, c.-à-d. que la commande **Configuration automate** ne sera accessible qu'après l'ouverture d'un projet.

Le configurateur est disponible en mode Hors ligne et En ligne.

Procédez ainsi lors de la configuration

Introduction

Cette section vous donne un aperçu général du processus de configuration.

Utiliser le menu de configuration

Certaines commandes doivent impérativement être exécutées et sond disponibles dans la fenêtre **Configuration automate**. Au départ, les commandes de menu masquées ne sont pas disponibles et peuvent être activées, pour l'extension de la configuration du matériel, dans le registre **Extensions de config.** peuvent être activés avec la commande **Sélectionner des extensions**.

Saisie des modules existants

Saisir manuellement les modules de l'API et comparer, en mode connecté, avec le matériel connecté. A l'issue de la saisie, les modules manquant dans Concept sont affichés dans la liste d'équipements E/S, ceci permettant de corriger à posteriori.

Procédez ensuite à l'adressage E/S individuel de chaque module.

Respectez, dans ce contexte, les références autorisées à utiliser :

Modules	**Références**
Modules analogiques d'entrée	Références 3x
Modules analogiques de sortie	Références 4x
Modules numériques d'entrée	Références 3x ou 1x
Module numérique de sortie	Références 4x ou 0x
Modules experts - Entrée	Références 3x ou 1x
Modules experts - Sortie	Références 4x ou 0x

Chargement de la configuration matérielle

La configuration matérielle d'un projet étant enregistrée, elle peut être chargée conjointement au programme de simulation Concept-SIM, Concept-SIM32 ou avec un automate. Ceci permet d'établir l'état IDENTIQUE entre l'appareil de programmation et API.

Note : Pour le transfert de configuration dans un API réel, désactivez Concept-SIM.

5.2 Configuration en mode local et connecté

Introduction

Aperçu

Cette section vous donne des informations sur la configuration en mode local et connecté

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Remarques générales	94
Fonctions disponibles en mode EN LIGNE et HORS LIGNE	95

Remarques générales

Introduction

En mode HORS LIGNE, aucune connexion n'est établie entre le terminal et API; la configuration peut être exécutée. En mode CONNECTÉ, la liaison étant établie entre le terminal et API, seule une configuration sous certaines réserves pourra être réalisée.

Fonctions disponibles en mode EN LIGNE et HORS LIGNE

Introduction

Cette section vous donne un aperçu des fonctions disponibles en mode local et connecté. Les possibilités disponibles en mode EN LIGNE sont différentes lors de l'utilisation du simulateur et de l'API réel.

Configuration en mode local

En mode HORS LIGNE, toutes les commandes de menu sont disponibles dans la fenêtre **Configuration automate** pour la configuration du matériel. Les sous-menus dans le répertoire **Extensions de config.** peuvent être activés, pour l'extension de configuration, dans le dialogue **Sélectionner des extensions**.

Si les API sont en mode connecté, passer en mode local à l'aide de la commande **En ligne** → **Déconnecter...**. Affichage de l'état NON CONNECT dans l'indicateur de la ligne d'état de la fenêtre Editeur.

Configuration en mode connecté et avec simulateur actif

Lorsqu'un simulateur ou une liaison Modbus est à l'état actif, aucune configuration n'est possible en mode EN LIGNE, ceci signifie qu'aune entrée ne peut être effectuée. Il est seulement possible de scruter et de lire les dialogues disponibles.

Passez en mode connecté à l'aide de la commande **En ligne** → **Connecter...**, ceci permettant d'établir la liaison entre l'appareil de programmation et API.

Configuration en mode EN LIGNE et API réel

En cas de liaison avec un API réel, la configuration est possible en mode EN LIGNE, dès que le niveau d'accès **Modifier la configuration** est activée.

Il n'est pas possible de configurer ou de reconfigurer un API se trouvant en mode FONCTIONNEMENT EN COURS. En présence d'un programme en cours d'exécution dans API, il faut arrêter le programme avant de poursuivre la reconfiguration. Arrêtez l'API à l'aide de **En ligne** → **Commande en ligne** → **Arrêter automate**. Après le traitement, les modifications sont transmises automatiquement au matériel au démarrage de l'API.

Note : Si, en mode EN LIGNE, vous effacez un groupe expert sur la liste d'équipement E/S, le chargeable correspondant est automatiquement effacé. Si vous souhaitez insérer ce groupe plus tard sur la liste d'équipement, un nouveau chargement est nécessaire.

Passez en mode connecté à l'aide de la commande **En ligne** → **Connecter...**, ceci permettant d'établir la liaison entre l'appareil de programmation et API.

Répercussion des modifications EN LIGNE

Si les conditions suivantes sont remplies, toutes les fenêtres animées se fermeront automatiquement, en cas de modification de la liste d'équipement E/S (par exemple effacement ou insertion d'un module.

Conditions

- Mode En ligne
- Section(s) animée(s)
- L'état entre l'API et l'appareil de programmation doit obligatoirement être EGAUX.
- API est arrêté
- Le niveau d'accès **Modifier la configuration** est activée

5.3 Configuration inconditionnelle

Introduction

Aperçu

Cette section décrit la configuration à exécuter de manière inconditionnelle et donne un aperçu des paramétrages préalables à définir dans le menu de configuration.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Condition préalable	98
Sélection automate	99
Sélection d'UC pour le type d'automate	100
Partition de mémoire API	105
Instructions chargeables	106
Ordonnanceur de segments	109
Affectation des E/S	111

Condition préalable

Introduction

C'est seulement après la sélection de l'UC dans le dialogue **Sélection automate** que toutes les autres commandes de menu sont disponibles dans la fenêtre **Configuration automate**.

Les boîtes de dialogue suivantes correspondent à une sélection minimale et sont obligatoirement à renseigner pour la configuration Hardware.

- Sélection automate
- Partition de mémoire API
- Instructions chargeables
- Ordonnanceur de segments
- Affectations d'E/S

Il est possible aussi de prendre en compte les préréglages, sous réserve que ceux-ci correspondent au Hardware utilisé.

Sélection automate

Introduction

Sélectionnez, en fonction de votre application, la gamme d'automate (Quantum, Compact, Momentum ou Atrium) et l'UC ainsi que la taille mémoire. Toutes les unités centrales disponibles sont indiquées dans la zone de liste.

Définir la zone de logique

En sélectionnant la gamme d'automate, vous pouvez étendre la zone de logique pour le langage de programmation voulu (CEI ou LL984) au type d'automate correspondant.

L'affectation et l'installation des instructions chargeables sont définies en fonction de la configuration suivante :

Sélection	Signification
Validée	Installation des instructions chargeables CEI. Un espace mémoire adapté peut être défini pour la zone CEI. L'affectation et l'installation des paires d'instructions chargeables sur l'UC sélectionnée se fait automatiquement dans la boîte de dialogue **Instructions chargeables**.
Invalider	Aucune installation d'instructions chargeables CEI. Ainsi, la zone CEI est entièrement désactivée et vous disposez alors de toute la zone de logique pour LL984.
984 seul/CEI seul	Quelques UC Momentum peuvent être programmées uniquement dans la zone CEI ou LL984.

Définir la mémoire totale CEI

En définissant la taille de la mémoire totale CEI et les données globales, vous définissez ainsi automatiquement lataille de la mémoire du programme CEI. En se basant sur cette taille, il est également possible de définir l'espace mémoire disponible pour le programme utilisateur LL984.

Note : Pour les données globales, il s'agit de l'espace mémoire des variables non localisées.

Note : Mémoire totale CEI = mémoire du programme CEI + données globales

Sélection d'UC pour le type d'automate

Introduction

Lors de l'installation de matériel (Concept EXECLoader), il vous est demandé de charger différents fichiers EXEC (*.BIN). Vous définissez ainsi les micrologiciels pour les différents types d'automate. Les tableaux suivants indiquent les types d'automate disponibles pouvant être utilisés en chargeant les fichiers EXEC avec les UC correspondantes.

Charger le micrologiciel pour les automates Quantum

Le tableau suivant indique les versions EXEC courantes incluses sur le CD Service Release et livrées avec Concept.

Type d'automate Quantum :

UC 140	Q186Vxxx (CEI+LL984)	Q486Vxxx (CEI+LL984)	Q58Vxxxx (CEI+LL984)	Q5RVxxxx (CEI+LL984)	QIECVxxx (CEI seulement) *	Mémoire CEI (Kilo-octet)
113 02	X (LL984 seulement)	-	-	-	-	
113 02S	-	-	-	-	X	150 maximum
113 02X	X (LL984 seulement)	-	-	-	-	
113 03	X	-	-	-	-	136 maximum
113 03S	-	-	-	-	X	379 maximum
113 03X	X	-	-	-	-	136 maximum
213 04	X	-	-	-	-	305 maximum
213 04S	-	-	-	-	X	610 maximum
213 04X	X	-	-	-	-	305 maximum
424 0x	-	X	-	-	-	465 maximum
424 0xX	-	X	-	-	-	465 maximum
434 12	-	-	X	-	-	890 maximum
534 14	-	-	X	-	-	2 550 maximum
434 12A (UC reconçue)	-	-	-	X	-	890 maximum
534 14A/B (UC reconçue)	-	-	-	X	-	2 550 maximum

Note : * Après avoir chargé le fichier EXEC QIECVxxx.BIN, il faut charger le fichier chargeable EMUQ.EXE dans le logiciel Concept dans la boîte de dialogue **Chageables**.

Charger le micrologiciel pour le fonctionnement en redondance d'UC LL984 Quantum	Les UC Quantum ne se terminant pas par X et S peuvent être utilisées pour un fonctionnement en redondance d'UC LL984. Pour cela, il faut charger un fichier EXEC spécial dans l'UC. Le système installe automatiquement l'instruction chargeable pour la redondance d'UC LL984 (CHS_208.DAT).
Charger le micrologiciel pour le fonctionnement en redondance d'UC CEI Quantum	Les UC 140, UC 434 12 et 140 UC 534 14 peuvent également être utilisées pour la redondance d'UC CEI. Pour cela, il faut charger un fichier EXEC spécial dans l'UC. Les instructions chargeables pour la redondance d'UC CEI (IHSB196.EXE et CHS_208.DAT) seront automatiquement installées par le système.
Charger le micrologiciel pour l'éditeur d'équation Quantum	Les UC Quantum ne se terminant pas par X et S peuvent être utilisées pour l'éditeur d'équation de LL984. Pour cela, un fichier EXEC spécial doit être chargé dans la mémoire flash de l'UC. Ce fichier EXEC n'est pas fourni avec Concept, mais est disponible sur Internet à l'adresse suivante : www.schneiderautomation.com.

Charger le micrologiciel pour les automates Momentum

Le tableau suivant indique les versions EXEC courantes incluses sur le CD Service Release et livrées avec Concept.

Type d'automate Momentum (UC 171 CBB 970 30) :

171 CBB	MPSV100.BIN (LL984 seulement)	MPSV100e.BIN (CEI seulement)	Mémoire CEI (Kilo-octet)
970 30-984	X	-	
970 30-CEI	-	X	236

Type d'automate Momentum (UC 171 CCC 7x0 x0) :

171 CCC	M1LLVxxx (LL984 seulement)	M1IVxxxE (CEI seulement)	Mémoire CEI (Kilo-octet)
760 10-984	X	-	
760 10-CEI	-	X	220
780 10-984	X	-	
780 10-CEI	-	X	220

Type d'automate Momentum (UC 171 CCC 9x0 x0) :

171 CCC	M1EVxxx (LL984 seulement)	M1EVxxxE (CEI seulement)	Mémoire CEI (Kilo-octet)
960 20-984	X	-	
960 30-984	X	-	
960 30-CEI	-	X	236
980 20-984	X	-	
980 30-984	X	-	
980 30-CEI	-	X	236

Type d'automate Momentum (UC 171 CCS 7x0 x0) :

171 CCS	M1LLVxxx (LL984 seulement)	M1IVxxxE (CEI seulement)	Mémoire CEI (Kilo-octet)
700 10	X	-	
700/780 00	X	-	
760 00-984	X	-	
760 00-CEI	-	X	160

Le fichier strippé EXEC de M1 prend en charge un maximum de 44 modules E/S.

Charger le micrologiciel pour les automates Compact

Pour toutes les UC Compact, charger le fichier EXEC **CTSXxxxD.BIN** dans la mémoire flash de l'UC.

Charger le micrologiciel pour les automates Atrium

Pour chaque UC Atrium, un fichier EXEC spécial (voir tableau ci-dessous) doit être chargé dans la mémoire flash de l'UC :

180 CCO	Fichier EXEC
121 01	AI3Vxxxx.BIN
241 01	AI5Vxxxx.BIN
241 11	AI5Vxxxx.BIN

Partition de mémoire API

Introduction

Pour l'élaboration de programme, prévoyez suffisamment de zones d'adresses pour le nombre requis de bits d'entrée, de sortie/bits d'annotation, mots d'entrée et de sortie/mots d'annotations .

Vous obtenez, au demeurant, un aperçu des valeurs de la bascule de verouillage :

- Bascule de verouillage max.
- Bascule de verouillage utilisée
- Taux d'occupation de la bascule de verouillage

Toute valeur non autorisée est signalée par un message d'erreur et peut être automatiquement adaptée à la valeur proposée

Données Standby Hot CEI

Après avoir configuré un système CEI Hot Standby, prévoyez suffisamment de zones d'adresses pour le nombre requis de mots d'entrée. Plus le nombre de mots d'entrée CEI Hot Standby est élevé, et plus la mémoire tampon de transfert, destiné à la partie CEI, est grande. Cela signifie que l'application CEI utilisée peut être, elle-aussi, d'autant plus grande.

⚠ ATTENTION

Influence du temps de cycle système !

La taille de la mémoire d'état configurée dans un projet de redondance d'UC CEI a une influence signifiante sur le temps de cycle système. Dès qu'un cycle logique est terminé, le suivant commence, une fois que toutes les données de mémoire d'état ont été transmises au module CHS.

Le non-respect de ces instructions peut provoquer des blessures ou des dommages matériels.

Instructions chargeables

Introduction

Les instructions chargeables sont des programmes à ne charger qu'en cas de besoin dans l'automate.

Les différentes utilisations des instructions chargeables sont abordées dans les sections suivantes.

Note : Lorsque vous supprimez en mode en ligne un module expert dans la liste d'affectation des E/S, l'instruction chargeable affectée est automatiquement supprimée. Si par la suite vous voulez de nouveau insérer ce module dans la liste d'affectation des E/S, un nouveau chargement est nécessaire.

Charger des instructions pour le système d'exécution CEI

Vous disposez des instructions chargeables suivantes pour une exécution combinée des programmes CEI et LL984 (UC 113 0x, UC 213 0x ou UC 424 02) :

Lorsque	Alors
vous désirez utiliser des UC avec processeur mathématique pour la programmation CEI,	vous installez la paire d'instructions chargeables @1S7196 et @2I7196.
vous désirez utiliser des UC sans processeur mathématique pour la programmation CEI,	vous installez la paire d'instructions chargeables @1SE196 et @2IE196.

Charger des instructions pour modules experts

Vous disposez des instructions chargeables suivantes pour les modules experts :

Lorsque	Alors
vous configurez le module 140 ESI 062 00 avec un système d'exécution 32 bits et le module 140-NOA-611-x0,	vous installez l'instruction chargeable ASUP196. **Note :** L'instruction chargeable ULEX196 est automatiquement installée. L'instruction chargeable ASUP196 n'est installée automatiquement qu'en cas d'utilisation d'UC 32 bits. En présence d'UC 16 bits avec fichier EXEC stripped (QIECVxxx.BIN), il est nécessaire d'installer l'instruction chargeable ASUP196 ultérieurement.
vous configurez le module 140 ESI 062 10,	vous installez la paire d'instructions chargeables NSUP + ESI. **Note :** Ces deux instructions chargeables ne font pas partie de l'ensemble logiciel Concept, mais sont livrées avec le module 140 ESI 062 10 et doivent être décompactées pour être installées (**Décompacter...**).

Charger des instructions pour LL984

Elles ne sont pas livrées avec Concept. Vous pouvez demander ces instructions chargeables via le "Automation Customer Service Bulletin Board (BBS)" (voir également le fichier README).

Charger des instructions pour la redondance d'UC

Vous disposez des instructions chargeables suivantes pour le fonctionnement en redondance :

Lorsque	Alors
vous utilisez le fonctionnement en redondance d'UC LL984,	l'instruction chargeable CHS_208 est automatiquement installée.
vous utilisez le fonctionnement en redondance d'UC CEI,	les instructions chargeables IHSB196 et CHS_208 sont automatiquement installées.

Charger des instructions utilisateur

Les instructions chargeables créées par les utilisateurs sont appelées instructions chargeables utilisateur (*.EXE, *.DAT). Elles se trouvent dans le répertoire Concept DAT et peuvent être installées, via le bouton de commande **Décompacter...**, dans la boîte de dialogue **Instructions chargeables**.

Charger des instructions pour la gestion CEI exclusive

Vous disposez des instructions chargeables suivantes pour la gestion CEI exclusive (UC 113 xxS sans processeur mathématique) :

Lorsque	Alors
votre application REAL utilise l'arithmétique,	vous installez l'instruction chargeable EMUQ196. **Note :** L'instruction chargeable est installée avec le fichier EXEC QIECVxxx (Installation dans Concept EXECLoader).

Chargement d'instructions pour l'INTERBUS et pour la gestion CEI uniquement

Vous disposez des instructions chargeables suivantes pour la gestion CEI :

Lorsque vous configurez l'UC...	Alors
• 113 02S • 113 03S • 213 04S • 534 14 • 434 12	vous installez l'instruction chargeable ASUP196. **Note :** L'instruction chargeable ULEX196 est automatiquement installée. L'instruction chargeable ASUP196 n'est installée automatiquement qu'en cas d'utilisation d'UC 32 bits. En présence d'UC 16 bits avec fichier EXEC stripped (QIECVxxx.BIN), il est nécessaire d'installer l'instruction chargeable ASUP196 ultérieurement.
113 03,	vous installez la paire d'instructions chargeables @1SE196 + @2IE196. L'instruction chargeable ULEX196 est automatiquement installée.
213 04,	vous installez la paire d'instructions chargeables @1S7196 + @2I7196. L'instruction chargeable ULEX196 est automatiquement installée.

Chargement d'instructions pour l'INTERBUS et pour la gestion LL984 uniquement

Vous disposez des instructions chargeables suivantes pour la gestion LL984 :

Lorsque vous configurez l'UC...	Alors
• 113 02 • 113 03 • 213 04	vous pouvez installer les instructions chargeables suivantes : • ULEX196 • @1S7196 + @2I7196 + ULEX196 **Note :** L'instruction chargeable ULEX196 est alors automatiquement installée.
• 534 14 • 434 12	les instructions chargeables ASUP196 et ULEX196 sont automatiquement installées.

Ordonnanceur de segments

Introduction

Si vous configurez une station E/S décentralisée (remote) (Drop), vous pouvez, dans la boîte de dialogue **Ordonnanceur de segments** spécifier la séquence et le mode de prise en charge des sections de LL984.

En cas d'extinction (dans la boîte de dialogue **Affectation des E/S**) d'une station E/S décentralisée configurée (Drop), elle est automatiquement évacuée dans le gestionnaire de segments.

Mode de fonctionnement

A chaque station E/S (Drop) est associé un segment. On ne peut dès lors pas introduire dans l'organisateur de segments moins de segments que ce qui est configuré dans l'équipement E/S des stations E/S (Drops). Dans l'organisateur de segments, le numéro maximal de segment est mis par défaut à 32.

Le configurateur vérifie la concordance entre les deux dialogues et ordonne les stations E/S (Drops) dans l'organisateur de segments. Vous êtes informés, via une fenêtre, de quelles stations E/S (Drops) ont été introduites.

Modification de la séquence de prise en charge des segments

La séquence de prise en charge des segments peut être modifiée par l'utilisateur, en éditant directement dans la ligne correspondante le numéro de segment ou le numéro de station E/S. Pour la station E/S locale (Drop), on introduit dans la première ligne de la boîte de dialogue dans les colonnes **E.-Stat.** et **A.-Stat.** automatiquement les **1**.

Si aucune séquence n'a été définie, les segments sont pris en charge dans l'ordre croissant.

Critères de tri pour les stations E/S introduites.

Les stations E/S nouvellement introduites (Drops) sont ordonnées selon les critères suivants dans le gestionnaire de segments :

Si...	Alors ...
Si une nouvelle station E/S est introduite,	alors elle est automatiquement rangée derrière la dernière ligne disponible.
Si tous les segments spécifiés ont déjà été utilisés,	le dernier segment est à nouveau utilisé pour l'entrée de la nouvelle station E/S (Drop) Càd qu'un numéro de segment peut se répéter, alors que les stations sont différenciées.

Modes disponibles de prise en charge des segments

Lors de l'installation du gestionnaire de segments, on peut sélectionner les modes de prise en charge suivants :

Type de prise en charge	Signification
De façon continue	Prise en charge cyclique
De façon contrôlée	Prise en charge commandée par l'utilisateur
WDT Reset	Reset (remise à zéro) minuterie chien de garde
Fin de logique	Fin de la prise en charge

Note : Au cas où on doit travailler dans LL984 avec des sous-programmes, le dernier segment configuré ne doit pas être traité dans le gestionnaire de segments. Le type de solution doit absolument être **Fin de logique**.

Autres installations dans le gestionnaire de segments

Pour le type de prise en charge "De façon contrôlée", seuls sont admissibles les numéros de référence 0x et 1x, qui spécifient quand la logique relative à la section correspondante est prise en charge

Les zones **E.-Stat.** et **A.-Stat.** autorisent l'entrée de numéros de stations E/S correspondants, qui doivent être configurés. Si vous entrez un **0**, on n'actionne aucune Entrée et Sortie pour ce numéro de segment.

Affectation des E/S

Introduction

Dans l'affectation des E/S vous configurez les stations d'E/S (Drops) avec les modules utilisés. Vous procédez ensuite à l'adressage E/S et au paramétrage des modules configurés.

Affectation des stations d'E/S

L'affectation du numéro des stations d'E/S peut être entreprise jusqu'au premier choix libre (de 2 à ...). Le premier numéro de station d'E/S est automatiquement reconnu comme la station d'E/S locale (Drop), et ne peut pas être édité.

Configurer l'extension d'embase

Le module 140 XBE 100 00 est nécessaire à l'extension d'embase. Vous pouvez ainsi monter une deuxième embase et vous obtenez 13 emplacements supplémentaires. Le module 140 XBE 100 00 se trouve sur les deux embases et nécessite une alimentation propre (alimentation secteur).

La configuration de l'embase étendue se fait dans Concept dans la première station d'E/S sur les emplacements 2-1 à 2-16.

Une description plus détaillée de la configuration d'une embase étendue avec le module 140 XBE 100 00 est présentée dans le chapitre *Backplane Expander Config, p. 127*.

⚠ ATTENTION

L'affectation des emplacements du 140 XBE 100 00 n'est pas affichée dans le configurateur si bien que des doubles affectations peuvent se produire.

Faites attention aux emplacements matériels du module et de l'alimentation secteur et n'affectez pas ces emplacements à d'autres modules de l'affectation d'E/S.

Le non-respect de ces instructions peut provoquer des blessures ou des dommages matériels.

Note : Le flux de données via une embase étendue est plus rapide que via un système à distance.

Affectation des zones d'E/S

Lors de l'affectation des zones d'E/S, les références suivantes sont admises :

- Références 3x pour les modules d'entrée analogiques
- Références 4x pour les modules de sortie analogiques
- Références 3x ou 1x pour les modules d'entrée ToR
- Références 4x ou 0x pour les modules de sortie ToR
- Références 3x ou 1x pour les modules expert (entrée)
- Références 0x ou 4x pour les modules expert (sortie)

Note : Une vérification de l'unicité de l'adressage a lieu, si bien qu'aucune double affectation d'adresse ne peut se produire au sein de la configuration.

Paramétrage

Les modules configurés peuvent être paramétrés séparément, afin de définir des configurations de variable selon le procédé.

Liaison à d'autres systèmes de réseau

En plus des stations d'E/S locales et décentralisées (Drops) il est possible d'établir avec des modules de connexion configurés des liaisons avec d'autres systèmes de réseau :

- Ethernet
- INTERBUS
- Profibus DP

Voir également le chapitre *Configuration de différents systèmes de réseau, p. 131* et *Exemples de configuration, p. 917*.

Lire la configuration E/S API

Les modules matériels sont exécutés en mode EN LIGNE avec l'API arrêté dans l'affectation d'E/S et peuvent être transférés comme suit :

Etape	Action
1	Ouvrez un projet.
2	Ouvrez la fenêtre **Configuration automate**.
3	Ouvrez avec la commande **Type de l'automate** la boîte de dialogue **Sélection automate** et sélectionnez le type de l'automate.
4	Connectez le PC avec l'automate (**En ligne** → **Connecter...**).
5	Ouvrez la boîte de dialogue **Affectation des E/S** (**Configuration Automate** → **Affectation des E/S**).
6	Ouvrez avec le bouton de commande **Editer...** la boîte de dialogue **Station d'E/S Quantum locale**.
7	Activez la case à cocher **Lire configuration E/S API**. **Réaction :** Le module concerné est identifié en couleur dans la colonne **Détecté**.
8	Cliquez deux fois sur les zones de texte en couleur de la colonne **Détecté**. **Réaction :** Les modules exécutés sont transférés dans la colonne **Module**.
9	Saisissez les zones d'adresse dans les colonnes correspondantes (**Déb.entr.**, **Fin E**, **Déb.sort.**, **Fin S**).
10	Après la convergence matérielle entre le PC et l'automate, vous pouvez passer à la configuration.

5.4 Configuration optionnelle

Introduction

Aperçu

Dans cette section, vous trouverez la description de la configuration à réaliser en option.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Configuration des informations ASCII.	116
Rendre disponibles d'autres fonctions dans le configurateur	117
Echange de données entre abonnés au réseau Modbus Plus	118
Nombre de mots vraiment utilisés lors de la réception de données (Peer Cop)	119
Protéger les données contre l'accès dans la mémoire de signal.	121
Paramétrage des interfaces	122
Options diverses	124

Configuration des informations ASCII.

Introduction

Pour la création des informations ASCII, on vous demande d'abord de spécifier un masque, qui contient le nombre des informations, la taille de la zone d'informations et les ports ASCII. Ensuite, vous pouvez créer les informations ASCII, qui sont alors traitées avec le langage de programmation Schéma à contacts.

Condition

Des informations ASCII ne sont possibles que dans la gamme Quantum, et ne peuvent être traitées qu'avec le langage de programmation LL984.

Procédure

Pour créer les informations ASCII, vous devez au préalable spécifier le masque :

Etape	Action
1	Ouvrez dans le menu principal **Configuration automate** → **ASCII** la boîte de dialogue **Installation ASCII**.
2	Indiquez dans la zone de texte **Nombre total de messages** une valeur de 1 - 999.
3	Indiquez dans la zonbe de texte **Taille zone messages** une valeur de 1 - 9999 octets.
4	Indiquez dans la zone de texte **Ports ASCII** une interface de 2 - 32.
5	Confirmez vos entrées avec le bouton de commande **OK**. **Réaction :** Les sélections sont enregistrées et la boîte de dialogue est fermée.
6	Ouvrez dans le menu principal **Projet** la boîte de dialogue **Editeur de messages ASCII** (avec la commande **Messages ASCII...**).
7	Créez ici les informations ASCII, voir à cet effet la description *Editeur de messages ASCII, p. 627*.

Rendre disponibles d'autres fonctions dans le configurateur

Introduction

Des fonctions supplémentaires peuvent être utilisées pour la configuration, si elles sont au préalable libérées ou installées dans le dialogue **Sélectionner extensions**.

Activer des fonctions/boîtes de dialogue étendues.

Par l'activation d'une case à cocher ou la définition de modules Ethernet, la commande de menu correspondante est validée dans la fenêtre **Configuration automate** → **ASCII** et peut être traitée.

Les fonctions/boîtes de dialogue suivantes peuvent être activées :

- Protection des données
- Diffusion des E/S
- Redondance d'UC
- Scrutateur E/S Ethernet

Note : Les fonctions disponibles dépendent de l'UC configurée. Voir à ce sujet l'aide en ligne « Sélectionner des extensions ».

Spécifier les modules couplés

Pour la liaison à d'autres systèmes de réseau, des modules couplés doivent être configurés. A cet effet, indiquez dans la zone de liste correspondante le nombre des modules, qui sont alors disponibles dans l'équipement E/S.

Les systèmes suivants peuvent être configurés :

- Ethernet TCP/IP
- Ethernet Symax
- Ethernet MMS
- Profibus DP

Note : Le nombre maximal des modules couplés dépend de l'UC configurée. Voir à ce sujet l'aide en ligne « Sélectionner des extensions ».

Echange de données entre abonnés au réseau Modbus Plus

Introduction

Lors d'une liaison Modbus Plus (MB+), vous pouvez configurer un API par la fonctionnalité de diffusion des E/S, d'une façon telle qu'un échange de données avec un autre API soit possible. En l'occurrence, la diffusion des E/S prend des données dans un domaine de référence à l'intérieur d'un API "source" et il place celles-ci par le réseau Modbus Plus (MB+) dans un domaine de référence déterminé d'un API "cible". Cette opération est exécutée de manière identique à chaque échange de jeton (rotation).

Par le Processeur Peer on peut recevoir des données d'entrée d'autres abonnés au réseau local dans le programme de l'utilisateur. De même, des données de sortie du programme de l'utilisateur peuvent être envoyées à d'autres abonnés au réseau local.

La Diffusion des E/S possède deux variantes pour l'échange de données :

- échange de données global
- échange de données spécifique

Condition

Cette commande **Diffusion des E/S** n'est disponible que si vous activez dans la boîte de dialogue **Sélectionner des extensions** la case à cocher **Diffusion des E/S**.

Echange de données global

Lors de l'échange de données global, les données envoyées par le API "source" actuel sont reçues par tous les appareils API "cibles" se trouvant dans le réseau Modbus Plus (MB+). Vous pouvez ainsi atteindre jusqu'à 64 appareils cibles, qui peuvent chacun recevoir les données à 8 adresses cibles de la Mémoire de signal.

Voir aussi section "*Nombre de mots vraiment utilisés lors de la réception de données (Peer Cop), p. 119*".

Echange de données spécifique

Lors de l'échange de données spécifique, des données émanant d'un API "source" sélectionné sont envoyées à un API "cible" sélectionné dans le réseau Modbus Plus (MB+). A cet effet, entrez dans une table a l'abonné source et cible correspondant (1-64) l'adresse respective pour l'échange de données.

L'adresse doit correspondre avec l'Adresse d'abonné MB+ au dos du module respectif. Cette configuration d'adresse est modifiable et doit être spécifiée avant l'équipement. (Voir aussi la description du matériel)

Selon la configuration du matériel, vous sélectionnez l'abonné, qui doit être lu ou écrit.

Nombre de mots vraiment utilisés lors de la réception de données (Peer Cop)

Introduction

Le nombre de mots utilisés ne doit pas dépasser 500. Pour l'éviter, une formule simple permet de calculer le nombre de mots utilisés.

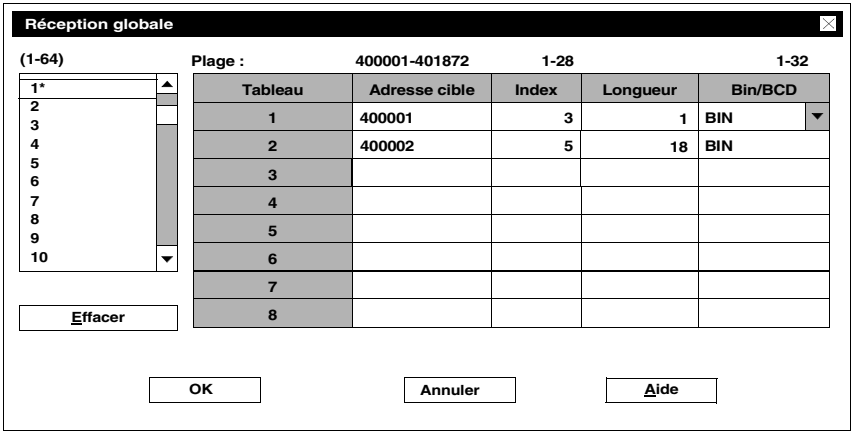
Formule

La formule permettant de calculer le nombre de mots utilisés est la suivante :

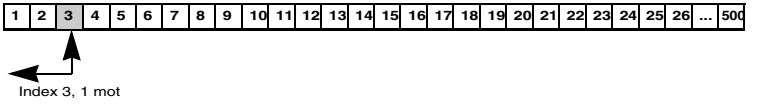
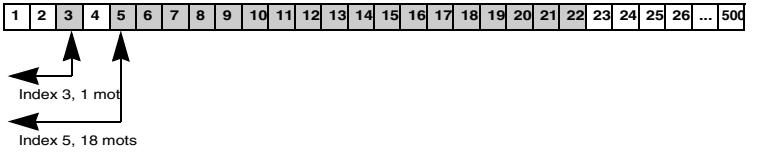
Longueur + (index - 1) = nombre de mots

Exemple

La boîte de dialogue Peer Cop **Réception globale** comporte l'entrée suivante :



Le processus suivant va se dérouler :

Etape	Action
1.	L'abonné au bus 1 envoie 1 mot à la référence de début de sous-zone 400001, commençant au niveau de l'index 3.
2.	Au niveau de l'index 3 (mot 3), la réception des données débute. (Les mots précédents interviennent dans le calcul.) Mot 1 à 500 
3.	La sous-zone 1 a besoin de 3 mots au total. Formule : 1 + (3 - 1) = 3
4.	L'abonné au bus 1 envoie 18 mots à la référence de début de sous-zone 400002, commençant au niveau de l'index 5.
5.	Au niveau de l'index 5 (mot 5), la réception des données débute. (Les mots précédents interviennent dans le calcul.) Mot 1 à 500 
6.	La sous-zone 2 a besoin de 22 mots au total. Formule : 18 + (5 - 1) = 22

Note : Seul le plus grand nombre de mots utilisés par abonné au bus est toujours considéré. Dans notre exemple, 22 mots sont utilisés au maximum sur 500 autorisés.
Pour plusieurs abonnés au bus, le plus grand nombre de mots utilisés par abonné au bus doit être ajouté.
Par exemple :

Abonné au bus 1 avec 22 mots
+
Abonné au bus 2 avec 28 mots

50 mots sur 500 mots autorisés.

Protéger les données contre l’accès dans la mémoire de signal.

Introduction

Les domaines d’adresse des sorties (bits de drapeau et mots de drapeau) peuvent être protégés si, dans la boîte de dialogue **Protection en écriture de la mémoire de données** vous spécifiez l’adresse, à partir de laquelle une écriture est possible. Toutes les adresses antérieures sont à écriture protégée.

Condition

La commande de menu **Protection d'accès aux données** n'est disponible que si vous activez dans le dialogue **Sélectionner extensions** la case à cocher **Protection des données** .

Entrée de la protection d’accès

Cette protection d’accès agit en liaison avec des accès aux données "normaux", qui se produisent en externe par une interface Modbus ou Modbus-Plus. Des accès à partir de l’appareil de programmation sont permis dans chaque cas, et évitent ce mécanisme de protection.

Paramétrage des interfaces

Introduction

Les interfaces suivantes doivent être paramétrées suivant leur utilisation dans Concept :

- Interface ASCII
- Interface Modbus

Paramétrage de l'interface ASCII

Pour une transmission d'informations ASCII vous pouvez, dans la boîte de dialogue **Sélections de port ASCII** spécifier les paramètres de communication série des interfaces de ports.

Note : La boîte de dialogue **Sélections de port ASCII** n'est disponible que si vous spécifiez d'abord, dans la boîte de dialogue **Setup ASCII** le nombre de ports ASCII.

Paramétrage de l'interface Modbus

Pour un couplage Modbus vous pouvez, dans la boîte de dialogue **Sélections de port Modbus** indiquer les paramètres de communication série de l'interface du port au terminal, à une UC et aux modules NOM (Network Option Module).

⚠ ATTENTION

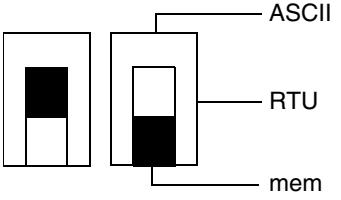
Ne pas opérer de modifications en ligne, cela entraînerait la fermeture de tous les éditeurs !

Les réglages des ports Modbus ne doivent pas être modifiés en mode En ligne, étant donné que tous les éditeurs sont fermés automatiquement.

Le non-respect de ces instructions peut provoquer des blessures ou des dommages matériels.

Note : Les réglages pour un couplage Modbus dans Concept n'ont d'effet que si la position de l'interrupteur à l'avant du module est dans la position la plus basse (mem).

Position de l'interrupteur sur le NOM



Note : Lorsque l'interrupteur de droite se trouve dans la position la plus élevée et celui de gauche est sur mem, le mode routeur est désactivé. Cela est valable à partir de la version 2.20 du micrologiciel. En fait, la liaison réseau entre Modbus et Modbus Plus est supprimée.

Paramétrage de l'interface lors de la liaison réseau entre Modbus et Modbus Plus

Vous pouvez établir une liaison réseau entre des participants Modbus et Modbus Plus dans la boîte de dialogue **Sélections de port Modbus** en activant la case à cocher **Mode Bridge**.

Note : Les réglages n'ont cependant d'effet que si l'interrupteur à l'avant du module est en position médiane (RTU).

Options diverses

Introduction

Dans la boîte de dialogue **Divers** vous pouvez configurer diverses options.

- Sortie pile
- Registre base de temps
- Indication de temps pour MMI-Applications (TOD)
- Autoriser bobines doubles
- Chien de garde (ms)
- Tranche de temps des modifications en ligne (ms)

Sortie pile

Vous pouvez indiquer une adresse d'un bit de sortie/mémento, qui affiche l'état de la pile. L'attribution sert à la surveillance de la pile. On peut ainsi procéder au remplacement en temps utile d'une pile faible, de sorte que l'on empêche la perte de données.

Registre base de temps

Le contenu du registre de temps est incrémenté toutes les 10 ms et il a une valeur libre entre 0000 et FFFF Hex.

Temps pour applications MMI (Date/Heure)

Cette indication de temps n'est prévue que pour une application MMI. Huit registres sont réservés pour le réglage de l'horloge.

L'indication TOD (Time of Day-Heure du Jour) se fait au format américain.

4xxxx	**Registre de commande**	
	Bit 1 (MSB) Bit 2 Bit 3 Bit 4	1 = Paramétrage des valeurs d'horloge 1 = Lecture valeurs d'horloge 1 = Bit de fin 1 = Bit d'erreur
4xxxx+1	Jour de la semaine (1 à 7)	
4xxxx+2	Mois (1 à 12)	
4xxxx+3	Jour (1 à 31)	
4xxxx+4	Année (00 - 99)	
4xxxx+5	Heures (0 à 23)	
4xxxx+6	Minutes (0 à 59)	
4xxxx+7	Secondes (0 à 59)	

Autoriser bobines doubles

Vous pouvez assigner plusieurs sorties à un bit de sortie/mémento. Pointez à cet effet la case à cocher, et indiquez dans la zone de texte **Première adresse de bobine :** la première adresse, à laquelle plusieurs sorties peuvent être assignées.

Note : Cette fonction n'est pas disponible pour la gamme API Momentum.

Chien de garde (ms*10) :

Vous pouvez spécifier une surveillance par impulsions du programme utilisateur, en entrant un nombre compris entre 2 et 255 (ms). Dès que les impulsions de comptage sont absentes pour un temps plus long que le temps défini, un message d'erreur apparaît.

Tranche de temps des modifications en ligne (ms)

Vous pouvez spécifier une surveillance temporelle de la communication entre les abonnés, en entrant un nombre compris entre 3 et 30 (ms). Dès que la communication dure plus longtemps que le temps prédéterminé, un message d'erreur apparaît.

5.5 Backplane Expander Config

Vue d'ensemble

Introduction

Ce chapitre décrit la fonction et la configuration de l'extenseur de processeur dorsal (backplane expander).

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Généralités relatives à l'extenseur d'arrière-plan	128
Modifier la répartition E/S	129
Traitement des erreurs	130

Généralités relatives à l'extenseur d'arrière-plan

Introduction	L'extenseur d'arrière-plan du Quantum fournit une extension de l'arrière-plan unique vers un drop local ou un drop RIO par le module 140 XBE 100 00.
Description de la fonction	Le module relie deux arrière-plans (primaire et secondaire) du Quantum à l'aide d'un câble personnalisé et gère toutes les communications de données entre les arrière-plans. Chaque arrière-plan requiert un module 140XBE10000 qui occupe un emplacement unique et sa propre alimentation.
Procédure lors d'une erreur	L'extenseur d'arrière-plan est conçu de sorte que, lorsqu'il n'est pas installé ou connecté correctement, il n'effectuera pas la fonctionnalité du rack primaire. C'est uniquement lorsque l'extenseur d'arrière-plan est installé et connecté correctement que les deux racks peuvent communiquer et être contrôlés par le régulateur de drop CPU ou RIO primaire.

Modifier la répartition E/S

Exigences

Actuellement, seuls les régulateurs Quantum gèrent l'extenseur d'arrière-plan. Le rack primaire contient le régulateur de drop CPU ou RIO et peut configurer tous les types de modules supplémentaires jusqu'à la limitation de l'adresse de l'emplacement physique. Tous les modules E/S peuvent également être ajoutés au rack secondaire. Cependant, les modules d'option, tels que les NOM, NOE et CHS doivent se trouver dans le rack primaire.

Pour placer un module dans le rack approprié, il est nécessaire d'ajouter un attribut supplémentaire dans la base de données du module E/S afin de spécifier que le module est uniquement disponible pour le rack primaire, secondaire ou pour les deux.

Configuration dans la répartition E/S

Le drop local du Quantum ou le drop RIO existants gèrent uniquement un rack ayant jusqu'à seize emplacements. Avec l'extenseur d'arrière-plan, il est étendu comme si le drop gèrait deux racks, disposant chacun de seize emplacements. En cliquant sur le bouton **...** de la colonne **Module**, tous les modules disponibles pour le rack sur lequel vous avez cliqué (primaire ou secondaire) seront affichés dans la boîte de dialogue de la sélection du module qui peut être sélectionné et attribué à l'emplacement courant.

Chaque rack requiert un module 140 XBE 100 00 afin que l'extenseur d'arrière-plan puisse fonctionner correctement.

Note : Le module 140 XBE 100 00 n'a pas un code de personnalité et il ne peut, par conséquent, être reconnu par Concept.

Le module ressemble à un emplacement vide dans la répartition E/S de Concept. Si un module est configuré dans le rack secondaire, l'utilisateur doit s'assurer qu'il y a un emplacement dans chaque rack réservé au module 140 XBE 100 00 et que tout le matériel est correctement connecté.

Traitement des erreurs

Introduction

Les procédés validés pour le rack primaire s'appliquent également au rack secondaire, tels que les références en double, les références d'entrée ou de sortie manquantes… En plus de la validation régulière existante, le cop trafic effectue des contrôles spéciaux pour l'extenseur d'arrière-plan.

Aucun emplacement réservé pour 140 XBE 1000 00

Si un module est présent dans le rack secondaire et qu'il n'y a pas d'emplacement vide dans aucun des deux racks lorsque l'utilisateur essaie de sortir de la boîte de dialogue de l'éditeur de racks, un message d'erreur s'affiche. "Il doit il y avoir un emplacement vide réservé au module 140 XBE 100 00 dans chaque rack afin que l'extenseur d'arrière-plan puisse fonctionner." La boîte de dialogue de l'éditeur de racks ne sera alors pas fermée.

Module spécial dans le rack secondaire

Pour éviter qu'un module spécial (tel que NOE, CHS…) soit ajouté au rack secondaire, la boîte de dialogue de l'éditeur de racks ne permet pas de couper/copier ces modules tête. Il contrôlera également les personnalités du module avant que l'utilisateur essaie d'effectuer une opération copier. Si un module non supporté du rack secondaire est présent, un message d'erreur s'affiche : " Le tampon contient des modules qui ne peuvent pas rester dans le rack secondaire." L'opération copier est arrêtée.

5.6 Configuration de différents systèmes de réseau

Introduction

Aperçu

Dans cette section, vous trouverez la description de la configuration de différents systèmes de réseau.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Configurer système INTERBUS	132
Configurer le système DP Profibus	133
Configurer Ethernet	135
Extension RTU	137
Scrutateur E/S / Ethernet	138
Comment utiliser le scrutateur E/S / Ethernet	142

Configurer système INTERBUS

Introduction

La configuration du système INTERBUS peut se faire à l'intérieur des familles API Quantum et Atrium.

Configuration de INTERBUSavec Quantum

Dans la famille Quantum, le couplage au Fernbus se fait dans une station E/S Quantum (Drop). A cet effet, le maître INTERBUS NOA 611 00 doit être configuré et paramétré dans CMD-Tool (Configuration Monitoring and Diagnostic Tool).

A cet effet, voir également le Exemple de configuration 4 (voir *Exemple Quantum - Commande INTERBUS, p. 949*).

Configuration de INTERBUS avec Atrium

Dans la famille Atrium, le couplage au Fernbus se fait par les blocs maîtres 180 CCO 121 01, 180 CCO 241 01 ou 180 CCO 241 11. Le maître INTERBUS CRP 660 0x est inséré automatiquement dans la station E/S locale (Drop). Les participants de la station E/S INTERBUS (Drop) sont configurés dans CMD-Tool (Configuration Monitoring and Diagnostic Tool), enregistrés comme fichier *.SVC et importés vers Concept. Après l'import dans le Equipement E/S la configuration peut être modifiée ultérieurement dans Concept.

A cet effet, voir également le Exemple de configuration 9 (voir *Exemple Atrium - Commande INTERBUS, p. 987*).

Configurer le système DP Profibus

Introduction

La configuration du système DP Profibus peut être effectuée avec les gammes d'automates Quantum et Atrium.

Configuration de DP Profibus avec Quantum

Dans la gamme Quantum, le couplage au système DP Profibus se fait dans une station E/S (Drop). Pour ce faire, vous devez tout d'abord indiquer le nombre de contrôleurs de bus (CRP-811-00) utilisés dans la boîte de dialogue **Sélectionner extensions**. Les modules apparaissent alors dans la zone de liste de la boîte de dialogue **Sélection de modules d'E/S** et peuvent être insérés dans l'Affectation des E/S.

La configuration des abonnés de DP Profibus s'effectue avec l'outil de configuration SyCon, elle est enregistrée dans un fichier *.CNF, puis transmise directement vers Concept. Cependant, la configuration (*.CNF) peut également être importée vers Concept ultérieurement.

⚠ ATTENTION

LES ADRESSES PROFIBUS DP PEUVENT ÊTRE ÉCRASÉES

Soyez attentif lors de la configuration de Profibus DP, car l'adressage sans espace de deux modules E/S 8 bits formant une limite de 16 bits n'est permis que lorsque ces deux modules 8 bits appartiennent au même maître DP Profibus. Si vous ne respectez pas cette consigne, les bits d'entrée d'un des modules (par ex. celui du maître Profibus DP A) pourraient être remplacés par ceux de l'autre module (par ex. celui du maître Profibus DP B).

Le non-respect de ces instructions peut provoquer des blessures ou des dommages matériels.

Import de la configuration DP Profibus

Pour importer la configuration (*.CNF) vers Concept, procédez comme suit :

Etape	Action
1	Ouvrez la boîte de dialogue **Affectation des E/S** à partir de la fenêtre **Configuration Automate**.
2	Sélectionnez la station E/S et ouvrez la boîte de dialogue **Station d'E/S Quantum locale** en cliquant sur le bouton de commande **Editer....**.
3	Double-cliquez sur la zone de texte **...** dans la colonne **Module**. **Effet :** la boîte de dialogue **Sélection de modules d'E/S** s'ouvre.
4	Sélectionnez le module CRP-811-00 dans la colonne **Adaptateur I/O**, et appuyez sur le bouton de commande **OK**. **Effet :** le module CRP-811-00 est ajouté à la liste des affectations d'E/S.
5	Sélectionnez la ligne du contrôleur de bus équipé (CRP-811-00) dans la boîte de dialogue **Station d'E/S Quantum locale**, et cliquez sur le bouton de commande **Paramètres**. **Effet :** la boîte de dialogue **CRP-811-00 (DP Profibus)** s'ouvre.
6	Le bouton de commande **Importer...** permet d'ouvrir la fenêtre **Sélection fichier source**.
7	Indiquez le chemin d'accès du fichier CNF à importer, et cliquez sur le bouton de commande **OK**. **Effet :** la configuration DP Profibus est insérée dans la liste d'affectation des E/S de Concept. **Remarque :** après l'insertion des abonnés DP Profibus dans Concept, les plages de référence de tous les modules et les données de diagnostic doivent être éditées.

Exemple de configuration

Vous trouverez un exemple de configuration sous Exemple 11 (voir *Exemple Quantum - Commande Profibus DP, p. 963*).

Configurer Ethernet

Introduction

Il est possible de configurer un système de bus Ethernet au sein des gammes d'automate suivantes.

- Quantum
- Atrium
- Momentum

Condition

Pour le couplage du système de bus Ethernet, une carte de réseau PCI doit être installée dans l'appareil de programmation. Ensuite, l'interface Ethernet doit être paramétré et les pilotes fournis sur CD doivent être installés (*Configurer Ethernet, p. 1008*), .

Après avoir affiché le bloc Ethernet dans la gestion de blocs centrale, vous pouvez vous faire attribuer par votre administrateur de réseau l'adresse Internet, le masque Subnet, la passerelle et le mode de transfert.

Configuration avec Quantum

Vous procédez comme suit pour la configuration Ethernet dans Concept :

Etape	Action
1	Ouvrez dans la fenêtre **Configuration automate** la boîte de dialogue **Sélectionner des extensions**.
2	Indiquez dans les zones de texte le nombre des blocs Ethernet (NOE). **Réaction :** Les modules apparaissent alors dans la zone de liste du dialogue **Sélection de modules d'E/S** et peuvent être insérés dans Affectation des E/S.
3	Ouvrez, dans la fenêtre **Configuration automate** le dialogue **Scrutateur E/S Ethernet**, dans lequel vous entrez les informations de l'administrateur de réseau (adresse Internet, masque Subnet, passerelle, mode de transfert).
4	Dans le menu principal **En ligne** ouvrez la boîte de dialogue **Connecter à l'API** (Commande **Connecter...**).
5	Dans la zone de liste **Type de protocole** sélectionnez l'option **TCP/IP**, et entrez dans la zone de texte **Adresse IP ou nom d'hôte DNS** l'adresse de la carte TCP/IP-Karte.
6	Après la programmation, ouvrez dans le menu principal **En ligne** la boîte de dialogue **Charger dans l'automate** (Commande **Chargement...**), et pressez le bouton de commande **Charger**. **Réaction :** Il apparaît un message, demandant si vous souhaitez démarrer la API.
7	Avant que vous confirmiez le message avec le bouton de commande **Oui** l'affichage "gauche" doit apparaître sur le bloc Ethernet.

Suppression de défaut

Ne démarrez la API qu'après la configuration, lorsque l'affichage "gauche" apparaît sur le bloc Ethernet. Si ce n'est pas le cas, retirez le bloc Ethernet hors de l'embase centrale et enfichez-le de nouveau. Si l'affichage "gauche" n'apparaît toujours pas, il doit y avoir un grave défaut.

Blocs Ethernet disponibles

Le nombre maximal des blocs NOE dépend de l'UC configurée (Sélection dans la boîte de dialogue **Sélection automate**) :

UC	Nombre des modules NOE
113 02/S/X	0 - 2
113 03/S/X	0 - 2
213 04/S/X	0 - 2
424 0x/X	0 - 6
434 12	0 - 6
534 14	0 - 6

Configuration avec Momentum

Une description de la configuration du système de bus Ethernet avec Momentum est donnée à la section Abschnitt *Exemple Momentum - Système de bus Ethernet, p. 1007*.

Extension RTU

Conditions

Pour que la commande de menu RTU soit disponible, vous devez sélection une UC Compact avec le langage de programmation LL984 dans le dialogue **Sélection automate**.

CTS-/RTS-Delay

Cette dialogue vous permet de régler le temps de retard pour CTS ou le RTS, indépendamment du port Comm 1 de votre API Compact. Cette caractéristique permet la communication par modem avec des radios demandant des trames de longue durée. La plage du temps de retard va de 0 à 500 ms, par unité de 10 ms. Entrez le temps de retard requis.

Secured Data Area (SDA)

Cette caractéristique vous permet de configurer, dans la RAM, une zone protégée contre la surimpression. Secured Data Area (SDA) est un module de la RAM de l'API Compact, configuré séparément comme espace de données 6x. Il n'est possible d'écrire sur le module SDA qu'à l'aide de fonctions spécifiques qui exigent une mémorisation protégée des données. Les commandes générales de Modbus, les fonctions intégrées, ne peuvent PAS écrire sur le module SDA. La fonction Lecture de Modbus (fonction 20) peut lire dans le module SDA, la fonction Ecriture de Modbus (fonction 21) ne peut pas écrire sur le module SDA. La taille du module SDA va de 0 à 128 K mots, en blocs d'une taille respective de 1 K mot seulement. Entrez la taille requise.

Référez-vous au manuel utilisateur correspondant, pour les fonctions spécifiques pour la taille requise du module SDA. Pour l'écoulement de gaz, par exemple, référez-vous au Guide de l'utilisateur « Lancement du bloc fonction chargeable Associates Gas Flow » (890 USE 137 00).

Protection de l'API par mot de passe

Pour la description de la protection par mot de passe, référez-vous à la section *Paramétrer mot de passe API, p. 680*.

Scrutateur E/S / Ethernet

Introduction

Cette fonction est disponible pour les modules Quantum suivants :
- 140-NOE-211-x0
- 140-NOE-251-x0
- 140-NOE-771-xx

Cette fonction est disponible pour les modules Momentum suivants :
- 171-CBB-970-30
- 171-CCC-960-20
- 171-CCC-980-20
- 171-CCC-980-30
- 171-CCC-960-30

Les paramètres d'adresse et du scanneur E/S Ethernet peuvent être modifiés à l'aide de la boîte de dialogue **Scrutateur d'E/S / Ethernet**. A partir de la fenêtre **Configuration automate** , sélectionner **Scrutateur d'E/S / Ethernet**. Cette option de menu n'est disponible que si vous avez sélectionné un adaptateur de processeur M1 avec un port Ethernet, ou si vous disposez de modules Quantum TCP/IP Ethernet (NOE) indiqués ci-dessus.

Cette section décrit comment configurer le port Ethernet, y compris l'adresse IP, d'autres paramètres d'adresse et l'exploration d'E/S.

Options de configuration Ethernet

L'écran du scanneur E/S / Ethernet offre trois options pour la configuration d'un adaptateur de processeur M1 :

Options d configuration	Signification
Spécifier l'adresse IP	Ceci représente l'option par défaut. Ceci vous permet d'entrer l'adresse IP, la passerelle et le masque Subnet dans les zones de texte au coin supérieur droit de l'écran.
Utiliser le serveur Bootp	Cliquez sur cette case d'option, si vous voulez assigner des paramètres d'adresse au serveur Bootp. Si vous sélectionnez cette option, les zones de texte des paramètres d'adresse, au coin supérieur droit de l'écran, seront marquées en couleur grise. Elles n'afficheront pas les paramètres d'adresse actuels.
Désactiver Ethernet	Cliquez sur cette case d'option, si vous voulez désactiver le port Ethernet. La désactivation du port pemet de réduire le temps de cycle pour l'adaptateur du processeur.

Définir les paramètres d'adresse Ethernet

Si vous voulez spécifier l'adresse IP, vous devez remplir toutes les quatre zones de texte placées au coin supérieur droit de la boîte de dialogue :

Paramètres	Signification
Adresse Internet	Entrez une adresse IP valide dans la zone de texte **Adresse Internet** (par exemple : 1.0.0.1). **Attention :** POSSIBILITE D'ADRESSES DOUBLES ! Demandez une adresse IP valide à votre administrateur de système, pour éviter des adresses doubles. **Failure to observe this precaution can result in injury or equipment damage.**
Multiplexeur	Consultez votre administrateur de système, pour déterminer la passerelle appropriée. Entrez-la dans la zone de texte **Multiplexeur**.
Masque Subnet	Consultez votre administrateur de système, pour obtenir le masque Subnet approprié. Entrez-le dans la zone de texte **Masque Subnet** (par exemple) : 255.255.255.0).
Type de transmission	Pour NOE, il existe un champ supplémentaire pour le type de transmission. Vos deux possibilités de sélection sont les suivantes : ETHERNET II ou IEEE 802.3

Configuration des E/S

Une fois, les paramètres d'adresse du port Ethernet ont été définis, vous pouvez affecter des paramètres pour l'exploration d'E/S.

La zone de texte **Module maître (emplacement)** contient le typ de module que vous avez configuré pour la communication Ethernet. En cas d'utilisation du contrôleur Ethernet de Momentum, l'emplacement sera toujours le numéro 1, et le type de module configuré est affiché dans la boîte de dialogue des variables Si vous configurez un NOE dans une baie standard, le numéro d'emplacement affecté dans l'affectation des E/S sera affiché avec le type de module. Cette zone de test affichera « Non affecté », jusqu'à ce que l'affectation des E/S soit terminée. En cas de configuration de plus d'un NOE, les paramètres d'exploration d'E/S se rapportent à l'unité actuellement dans la boîte de dialogue, à partir de laquelle vous pouvez sélectionner une unité additionnelle en activant la liste déroulante.

La zone de texte **Bloc de santé (1x/3x)** n'est disponible que si vous utilisez 140-NOE-771-xx. Le timeout défaut utilisé pour placer le bit de fonctionnement. Si la réponse arrive avant la fin de la période du timeout défaut, le bit de fonctionnement est placé ; le cas contraire, le bit est supprimé. Si le timeout défaut est à zéro, le bit de fonctionnement est placé sur VRAI dès que la communication a été établie, et il n'est jamais effacé.

Note : Pour la configuration du bloc de santé, référez-vous dans le manuel d'utilisateur Modules Ethernet Quantum NOE 771 xx, mod èle n° 840 USE 116 00.

La zone de texte **Bloc de diagnostic (3x/4x)** n'est disponible que si vous utilisez Ethernet Momentum (M1E), et vous permet de définir le registre de départ d'un certain nombre de bits utilisés pour le diagnostic. Le module peut être spécifié en registres 3x ou 4x. Pour plus d'informations, référez-vous au manuel d'utilisateur Modules Ethernet Quantum NOE 771 xx , modèle n 840 USE 116 00.

Tableau de configuration du scanneur E/S

Colonne	Description
Adresse IP esclave	Entrez l'adresse IP du module esclave dans la colonne (par exemple : 128.7.32.54). Cette adresse sera enregistrée dans un menu déroulant, et vous pourrez donc l'utiliser sur une autre ligne en cliquant sur la flèche de direction vers le bas et en la sélectionnant.
ID unité	Si le module esclave est un dispositif E/S connecté au module esclave spécifié, utilisez la colonne d'ID d'unité, pour indiquer le numéro du dispositif. L'ID unité est utilisé avec le pont de Modbus Plus à Ethernet, pour l'acheminement vers les réseaux Modbus.
Timeout défaut	Utilisez cette colonne, pour spécifier la durée en ms de tentative de transaction avant le timeout. Les valeurs valables sont comprises dans la plage de 0 à 50 000 ms (1 min.). Pour le timeout, entrez 0.

Colonne	Description
Taux de rép.	Utilisez cette colonne, pour spécifier la fréquence répétition de la transaction en ms. Les valeurs valables sont comprises dans la plage de 0 à 50 000 ms (1 min.). Pour répéter la transaction continuellement, entrez 0.
Lecture maître réf.	Utiliser la fonction de lecture, pour lire des données de l'esclave vers le maître. Cette colonne spécifie la première adresse à lire (par exemple : 400001).
Lecture esclave réf.	Utiliser la fonction de lecture, pour transférer des données de l'esclave au maître. Cette colonne spécifie la première adresse à lire, jusqu'à 125 (par exemple : 400050).
Longueur de lecture	Utiliser la fonction de lecture, pour lire des données de l'esclave vers le maître. Cette colonne spécifie le nombre de registres à lire (par exemple : 20).
Ecriture maître réf.	Utiliser la fonction d'écriture, pour transférer des données du maître à l'esclave. Cette colonne spécifie la première adresse à écrire (par exemple : 400100).
Ecriture esclave réf.	Utiliser la fonction d'écriture, pour transférer des données du maître à l'esclave. Cette colonne spécifie la première adresse à écrire, jusqu'à 100 (par exemple : 400040).
Longueur d'écriture	Utiliser la fonction d'écriture, pour transférer des données du maître à l'esclave. Cette colonne spécifie le nombre de registres à écrire (par exemple : 40).
Description	Vous pouvez entrer une courte description (jusqu'à 32 caractères) de la transaction dans cette colonne.

Note : Vous pouvez insérer des commande de lecture et d'écriture sur la même ligne.

Comment utiliser

Pour plus d'informations sur l'utilisation du dialogue du scanneur E/S / Ethernet, voir à la section *Comment utiliser le scrutateur E/S / Ethernet, p. 142*.

Comment utiliser le scrutateur E/S / Ethernet

Introduction

Cette section décrit comment procéder à la configuration de votre scrutateur d'E/S Ethernet à l'aide des boutons **Copier**, **Couper**, **Insérer**, **Effacer** et **Remplir vers le bas**.

Copier et insérer

Pour gagner du temps lors de la saisie de commandes similaires de lecture et d'écriture, vous pouvez copier et insérer des lignes entières dans votre configuration :

Etape	Action
1	Sélectionnez la ligne que vous voulez copier en cliquant sur le numéro de la ligne à l'extrême gauche.
2	Cliquez sur le bouton **Copier** au-dessus de la liste de configuration d'E/S.
3	Sélectionnez la ligne sur laquelle vous voulez insérer les données (en cliquant sur le numéro de la ligne à l'extrême gauche).
4	Cliquez sur le bouton **Insérer**.

Couper et insérer

Pour déplacer une ligne dans votre liste de configuration, procédez comme suit :

Etape	Action
1	Sélectionnez la ligne que vous voulez déplacer en cliquant sur le numéro de la ligne à l'extrême gauche.
2	Cliquez sur le bouton **Couper** au-dessus de la liste de configuration d'E/S.
3	Sélectionnez la ligne sur laquelle vous voulez insérer les données (en cliquant sur le numéro de la ligne à l'extrême gauche).
4	Cliquez sur le bouton **Insérer**. **Remarque :** Des lignes multiples peuvent être coupées/copiées et insérées. Le nombre de lignes effectivement insérées est limité par le nombre de lignes sélectionnées. Par exemple, si vous copiez 10 lignes dans le presse-papiers et sélectionnez une zone de 6 lignes à insérer, seules les six premières lignes contenues dans le presse-papiers seront insérées.

Supprimer

Pour effacer une ligne dans la liste de configuration, procédez comme suit :

Etape	Action
1	Sélectionnez la ligne que vous voulez effacer en cliquant sur le numéro de la ligne à l'extrême gauche.
2	Cliquez sur le bouton **Effacer** au-dessus de la liste de configuration d'E/S. **Remarque :** Vous effacez plusieurs lignes.

Remplir vers le bas

Pour copier une partie de toute ligne sur la ligne suivante ou sur une série de lignes contiguës, utilisez le bouton **Remplir vers le bas**, en suivant les étapes indiquées dans le tableau :

Etape	Action
1	Utilisez votre souris pour sélectionner les données que vous voulez copier et les cellules dans lesquelles vous souhaitez les copier. **Remarque :** Vous devez sélectionner un bloc de cellules contiguës, les données à copier se trouvant sur la première ligne. Vous ne pouvez pas sélectionner deux blocs séparés.
2	Cliquez sur le bouton **Remplir vers le bas**. **Résultat :** Les données de la première ligne sont copiées dans les cellules sélectionnées dans le bloc défini.

Modules Ethernet NOE

Dans cette boîte de dialogue, vous pouvez paramétrer les modules Ethernet NOE 140 NOE 211 x0,140 NOE 251 x0 et 140 NOE 771 10 (dans la zone **Configuration Ethernet**).

Dans cette boîte de dialogue, vous pouvez paramétrer le module 140 NOE 771 00 et réaliser son adressage (dans la zone **Configuration scrutateur d'E/S**).

Vous recevez une description de la fonction pour les modules suivants :

- 140 NOE 211 x0
- 140 NOE 251 x0
- 140 NOE 771 xx

Modules Momentum Ethernet

Dans cette boîte de dialogue, vous pouvez réaliser l'adressage des modules Momentum Ethernet (dans la zone **Configuration scrutateur d'E/S**).

Vous recevez une description de la fonction pour les modules suivants :

- 171 CBB 970 30 CEI
- 171 CBB 970 30 984
- 171 CCC 980 30 CEI
- 171 CCC 980 30 984
- 171 CCC 980 20 984
- 171 CCC 960 30 CEI
- 171 CCC 960 30 984
- 171 CCC 960 20 984

5.7 Paramètres de sécurité Quantum dans le configurateur

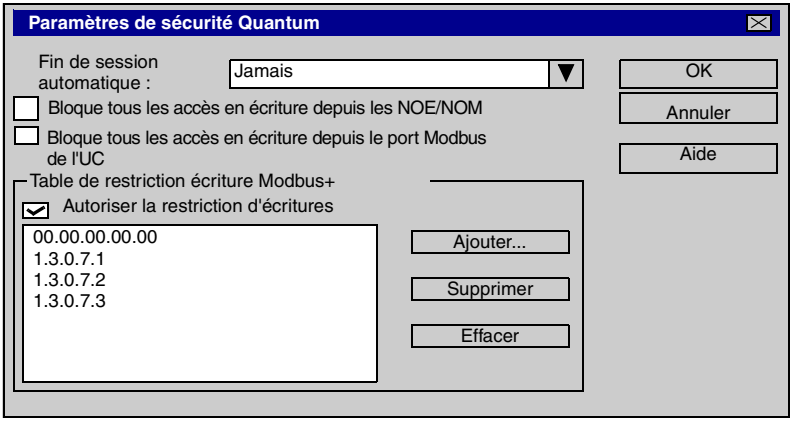
Paramètres de sécurité Quantum

Introduction

Avec la configuration des UC Quantum 140 434 12A et 140 534 14A/B, vous pouvez définir certains paramètres de sécurité qui apparaissent dans le fichier journal *.LOG. Cela permet de garantir une documentation sûre des processus contenant les enregistrements de fins de session automatiques, d'accès en écriture depuis les NOE/NOM à l'API et le nombre d'utilisateurs restreints (12 au maximum) pour l'accès réseau en écriture.

Les paramètres de sécurité sont définis dans la boîte de dialogue **Configurer** → **Extension de sécurité**.

Dialogue **Paramètres de sécurité Quantum**:



Condition

Les paramètres de sécurité ne sont disponibles que si les conditions suivantes sont remplies :

- Droits de superviseur (voir dans Concept sous **Aide** → **A propos...** → **Utilisateur actuel :**)
- uniquement pour les UC 140 CPU 434 12A et 140 CPU 534 14A/B

Fin de session automatique

La fin de session automatique permet de fermer la session utilisateur en l'absence d'activité sur cette liaison pendant une durée prédéfinie (max. 90 minutes). Il peut s'agir par ex. de l'absence d'accès en lecture/écriture à l'API de la part du PC.

La valeur **Jamais** bloque la fonction, c.-à-d. que la fin de session n'est pas automatique.

Note : La fin de session automatique ne fonctionne pas lorsque :
- le PC (Concept) n'est pas relié à l'API via le port Modbus Plus local de l'UC mais via un module NOE/NOM
  et
- que sur ce même module NOE/NOM, un autre équipement qui accède en lecture à l'API est branché.

Bloque tous les accès en écriture depuis les NOE/ NOM

Avec le blocage de toutes les écritures depuis les modules

- 140 NBE 210 00 (Code d'ID 0x0406)
- 140 NBE 250 00 (Code d'ID 0x0407)
- 140 NOE 211 00 (Code d'ID 0x0404)
- 140 NOE 251 00 (Code d'ID 0x0405)
- 140 NOE 311 00 (Code d'ID 0x0408)
- 140 NOE 351 00 (Code d'ID 0x0409)
- 140 NOE 511 00 (Code d'ID 0x040A)
- 140 NOE 551 00 (Code d'ID 0x040B)
- 140 NOE 771 00 (Code d'ID 0x040D)
- 140 NOE 771 01 (Code d'ID 0x0422)
- 140 NOE 771 10 (Code d'ID 0x040E)
- 140 NOE 771 11 (Code d'ID 0x0423)
- 140 NOM 211 00 (Code d'ID 0x010C)
- 140 NOM 212 00 (Code d'ID 0x010C)
- 140 NOM 252 00 (Code d'ID 0x010C)
- 140 NWM 100 00 (Code d'ID 0x0420)

sur l'API, l'UC ignore toutes les commandes d'écriture et les refuse en retournant un message d'erreur.

Note : Si la case **Bloque tous les accès en écriture depuis les NOE/NOM** est cochée, les opérations de lecture du MSTR ne sont pas exécutées. (L'état d'erreur du module MSTR n'indique pourtant pas d'erreur !)

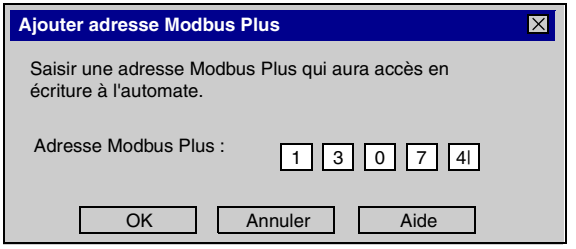
Bloquer tous les accès en écriture depuis le port Modbus de l'UC

Pour bloquer l'accès en écriture des ports Modbus de l'UC Quantum, activez la case à cocher **Bloquer tous les accès en écriture depuis le port Modbus de l'UC**.

Accès en écriture limité pour réseau Modbus Plus

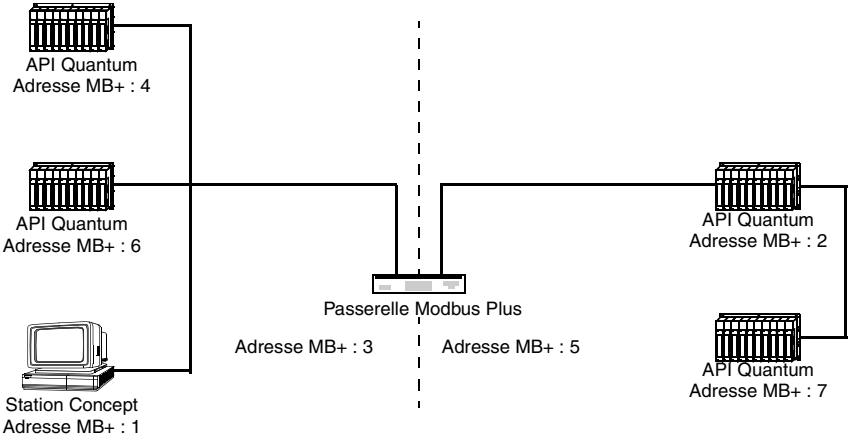
Pour le réseau Modbus Plus, il est possible de définir un nombre limité d'abonnés ayant accès à l'API. Un total maximum de 12 abonnés est autorisé. L'adresse d'abonné du PC est inscrite automatiquement dans la liste des abonnés et ne peut pas être supprimée.

Dialogue **Ajouter adresse Modbus Plus** (cliquer sur **Ajouter...**)



Exemples de chemins Modbus Plus

Réseau Modbus Plus :



L'adresse doit être saisie du point de vue de l'API prêt à recevoir au niveau de l'émetteur et débute de ce fait par la première passerelle ou l'API suivant. Le fait que l'abonné émetteur et l'abonné récepteur fassent partie du même segment Modbus Plus (pas de ponts/passerelles) ou non (séparés par un ou plusieurs ponts/passerelles) joue un rôle déterminant.

Exemple 1 :

Concept (adresse 1 MB+) écrit à l'API 6. Il n'y a ni pont ni passerelle entre les deux abonnés. L'adresse entrée est la suivante : 1 ou 1.0.0.0.0

Exemple 2 :

L'API 2 (adresse 2 MB+) écrit à l'API 6. Les deux abonnés sont séparés par une passerelle (adresse 3 MB+). L'adresse entrée est la suivante : 3.2.0.0.0

Note : Seule la première adresse Modbus Plus peut être reconnue par l'API. Ainsi, lorsque cette première adresse est un pont ou une passerelle, tous les appareils du réseau situés derrière ce pont ou cette passerelle ont alors un accès en écriture à l'API. Dans notre exemple, l'API 7 pourrait donc également écrire à l'API 6 (adresse : 3.7.0.0.0).

Structure schématique de la mémoire de l'API et optmisation de la mémoire.

6

Introduction

Aperçu

Ce chapitre décrit la structure schématique de la mémoire de l'API et l'optimisation de cette mémoire pour les différentes familles d'API.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
6.1	Structure schématique de la mémoire de l'API	151
6.2	Remarques générales sur l'optimisation de la mémoire	153
6.3	Optimiser la mémoire pour l'UC X13 0X et 424 02 Quantum	159
6.4	Optimisation de la mémoire pour l'UC 434 12(A) et 534 14(A/B) Quantum	173
6.5	Optimisation de mémoire pour les UC Compact	185
6.6	Optimisation de mémoire pour les UC Momentum	197
6.7	Optimisation de mémoire pour les UC Atrium	205

6.1 Structure schématique de la mémoire de l'API

Structure par principe de la mémoire de l'API

Introduction

La mémoire d'une API comprend en principe trois parties :
- la mémoire pour le fichier EXEC,
- la mémoire d'état et
- la mémoire du programme.

La mémoire pour le fichier EXEC

Le fichier EXEC comprend le système d'exploitation et un ou deux systèmes d'exécution (CEI et/ou LL984) servant au déroulement du programme utilisateur.

Mémoire d'état

La mémoire d'état peut être divisée en différentes zones :
- les références 0x, 1x, 3x et 4x utilisées,
- une réserve pour des références 0x, 1x, 3x et 4x supplémentaires,
- éventuellement une zone de mémoire étendue pour les références 6x.

Mémoire du programme

La mémoire du programme peut être divisée en différentes zones :
- l'affectation des E/S, etc.,
- une réserve pour extensions,
- les messages ASCII (si utilisés), la configuration de diffusion des E/S (si utilisée), la configuration Ethernet (si utilisée) etc.,
- une réserve pour extensions,
- les instructions chargeables CEI (si nécessaires),
- les données globales, se composant des variables non localisées,
- la mémoire du programme CEI avec les codes de programme, les codes EFB et les données de programme (données de section et données d'instance DFB).
- éventuellement l'instruction chargeable ULEX pour INTERBUS ou d'autres instructions chargeables,
- la mémoire du programme LL984.

6.2 Remarques générales sur l'optimisation de la mémoire

Introduction

Vue d'ensemble

Ce chapitre décrit les généralités sur l'optimisation de la mémoire

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Possibilités d'optimisation de la mémoire	154
Ne dépendant pas de l'automate	155

Possibilités d'optimisation de la mémoire

Description

Les possibilités d'optimisation de la mémoire dépendent en partie de la gamme d'automate et de l'UC utilisées :

- *Ne dépendant pas de l'automate, p. 155*
- *Optimiser la mémoire pour l'UC X13 0X et 424 02 Quantum, p. 159*
- *Optimisation de la mémoire pour l'UC 434 12(A) et 534 14(A/B) Quantum, p. 173*
- *Optimisation de mémoire pour les UC Compact, p. 185*
- *Optimisation de mémoire pour les UC Momentum, p. 197*
- *Optimisation de mémoire pour les UC Atrium, p. 205*

Ne dépendant pas de l'automate

Introduction

Il existe 3 possibilités d'optimisation de la mémoire ne dépendant pas de l'automate :

- *Optimiser la mémoire d'état pour les références 0x et 1x, p. 156*
- *Ne charger que les instructions chargeables attendues, p. 157*
- *Optimiser la réserve pour extensions, p. 157*

Optimiser la mémoire d'état pour les références 0x et 1x

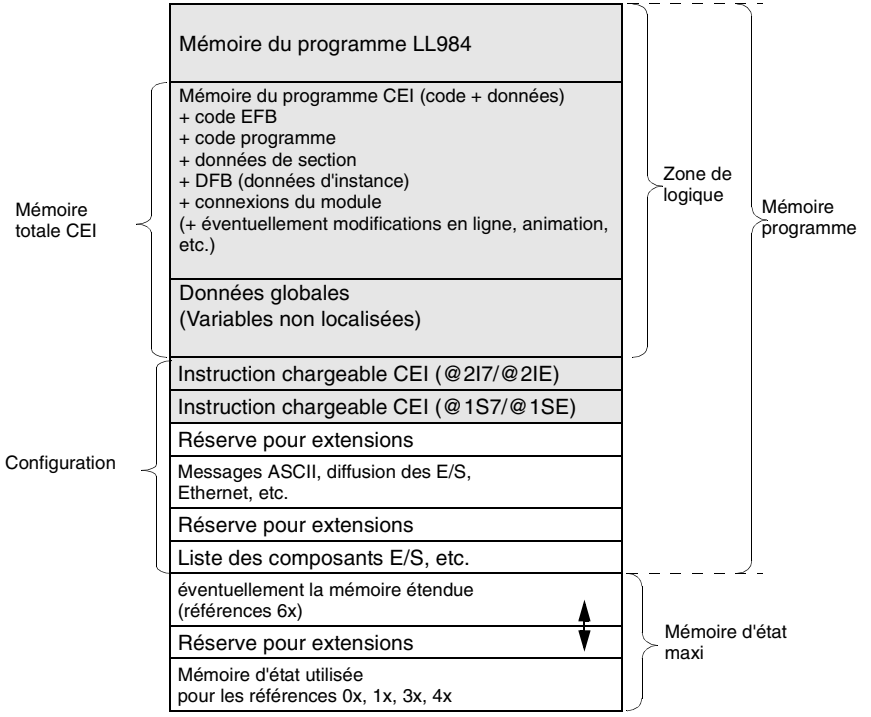
La mémoire d'état comprend les valeurs actuelles des références 0x, 1x, 3x et 4x.

Même lorsque la zone de la mémoire d'état est en dehors de la zone de la mémoire du programme, la taille de la mémoire d'état pour les références 0x et 1x influence la taille de la mémoire du programme. Ne sélectionnez donc pas une taille trop importante pour la zone de mémoire d'état destinée à ces références. Théoriquement, vous avez besoin d'autant de références 0x et 1x que le nombre nécessaire pour votre matériel. Vous avez cependant besoin d'un nombre légèrement supérieur de références si vous voulez étendre votre affectation des E/S. Pendant la phase de création du programme utilisateur il est conseillé de définir un nombre largement suffisant de références, même si celles-ci seront changées ultérieurement. A la fin de la phase de programmation, le nombre de références peut alors être réduit afin de laisser davantage de place à disposition du programme utilisateur.

La configuration des références 0x et 1x se trouve dans **Projet** → **Configuration automate** → **Partition mémoire de l'API**.

Cette boîte de dialogue vous donne un aperçu de la taille de la zone de la mémoire d'état occupée et du pourcentage maximum de la mémoire d'état que cela représente.

Optimiser la mémoire d'état pour les références 0x, 1x, 3x et 4x :



Ne charger que les instructions chargeables attendues

Toutes les instructions chargeables que vous avez installées sont chargées dans la zone de la mémoire du programme et y occuperont donc de l'espace. N'installez donc que les instructions chargeables dont vous avez vraiment besoin (voir également *Instructions chargeables, p. 106*).

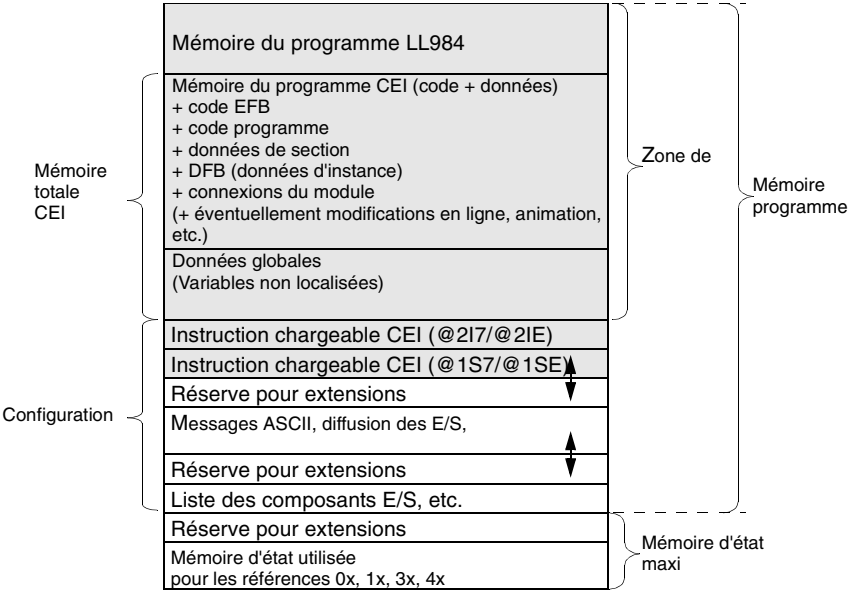
L'espace mémoire occupé par les instructions chargeables installées est affiché dans la boîte de dialogue **Instructions chargeables** sous **Octets utilités** (**Projet** → **Configurateur automate**). Cette valeur est calculée à partir de la taille des fichiers des instructions chargeables et de la mémoire qui a été réservée pour les instructions chargeables.

Optimiser la réserve pour extensions

Il est possible de réserver de l'espace mémoire pour des extensions futures dans la zone d'affectation (Affectation des E/S) et dans la zone des extensions de configuration (diffusion des E/S). Cette espace mémoire est nécessaire si par ex. les données d'affectation ou de diffusion des E/S doivent être modifiées en ligne. Pendant la phase de création du programme utilisateur il est conseillé de définir des réserves largement suffisantes, même si par la suite il y a des changements fréquents. A la fin de la phase de programmation, les réserves peuvent alors être réduites afin de laisser davantage de place à disposition du programme utilisateur.

La configuration des réserves pour affectation se trouve sous **Projet** → **Configuration automate** → **Affectation des E/S** → **Réserve pour extension**. La configuration pour les réserves destinées à la diffusion des E/S se trouve sous **Projet** → **Configuration automate** → **Extensions de config.** → **Sélectionner extensions** → **Diffusion des E/S** → **Réserve pour extension**.

Optimiser la réserve pour extensions



6.3 Optimiser la mémoire pour l'UC X13 0X et 424 02 Quantum

Introduction

Vue d'ensemble

Cette section décrit l'optimisation de la mémoire pour les UC X13 0X et 424 02 Quantum.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Généralités sur l'optimisation de la mémoire pour l'UC X13 0X et 424 02 Quantum	160
Sélectionner le fichier EXEC optimal	162
Utiliser la mémoire étendue (mémoire d'état pour les références 6x)	166
Synchronisation de la zone CEI et de la zone LL984	168
Synchronisation des zones pour les données globales et la mémoire du programme CEI	170

Généralités sur l'optimisation de la mémoire pour l'UC X13 0X et 424 02 Quantum

Mémoire logique

La zone de la mémoire du programme qui accueille votre programme utilisateur est appelée zone de logique. Cette zone détermine également la taille maximale de votre programme utilisateur.

La taille actuelle de la zone de logique peut être visualisée sous **Projets** → **Configuration automate** dans la vue d'ensemble de la configuration dans la zone **Automate**. La taille de la mémoire est affichée en nodules pour LL984(1 nodule équivaut à 11 octets) et en ko pour CEI.

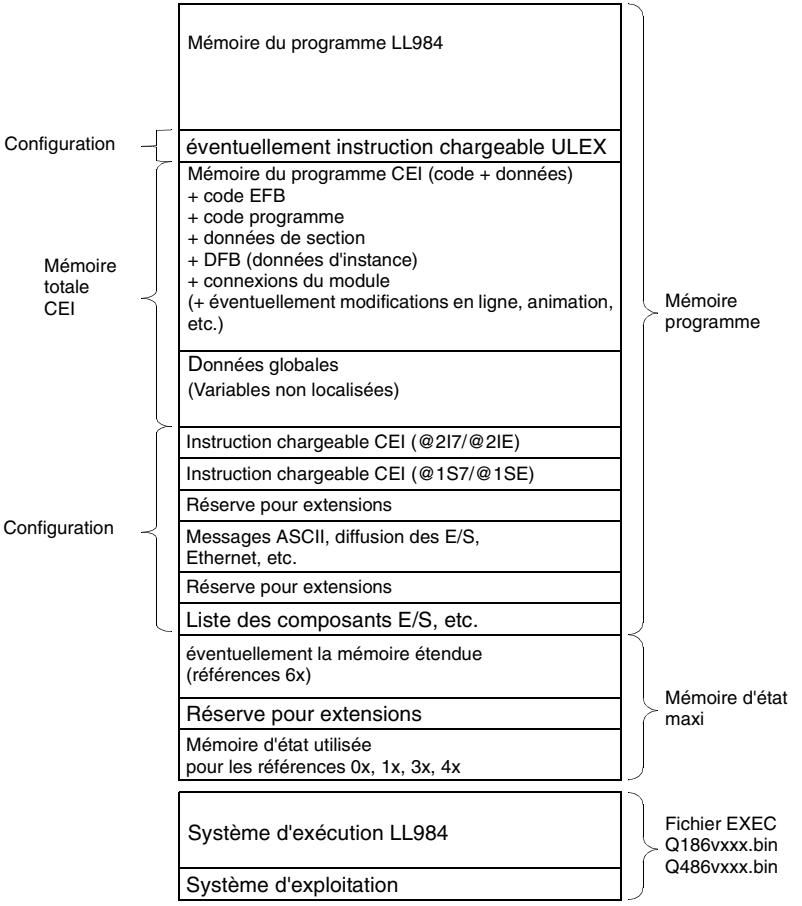
Optimisation de la mémoire logique

Vous avez différentes possibilités pour optimiser la mémoire logique en fonction de vos besoins :

- *Sélectionner le fichier EXEC optimal, p. 162*
- *Utiliser la mémoire étendue (mémoire d'état pour les références 6x), p. 166*
- *Synchronisation de la zone CEI et de la zone LL984, p. 168*
- *Synchronisation de la zone CEI et de la zone LL984, p. 168*

Note : Prenez également en compteles possibilités d'optimisation de la mémoire indépendantes de l'automate (voir *Remarques générales sur l'optimisation de la mémoire, p. 153*).

Structure de la mémoire de l'UC X13 0X (représentation simplifiée) :



Sélectionner le fichier EXEC optimal

Introduction

La possibilité principale qui est également la plus simple consiste à charger le fichier EXEC optimal pour vos besoins dans l'automate (voir également *Introduction à l'installation*).

En fonction du fichier EXEC choisi, des zones sont réservées dans la mémoire du programme de l'automate pour CEI et/ou LL984. Si vous installez un 'fichier EXEC combiné' et qu'ensuite vous n'utilisez qu'un seul des deux types de langage dans votre programme utilisateur, la mémoire du programme n'est pas utilisée de manière optimale.

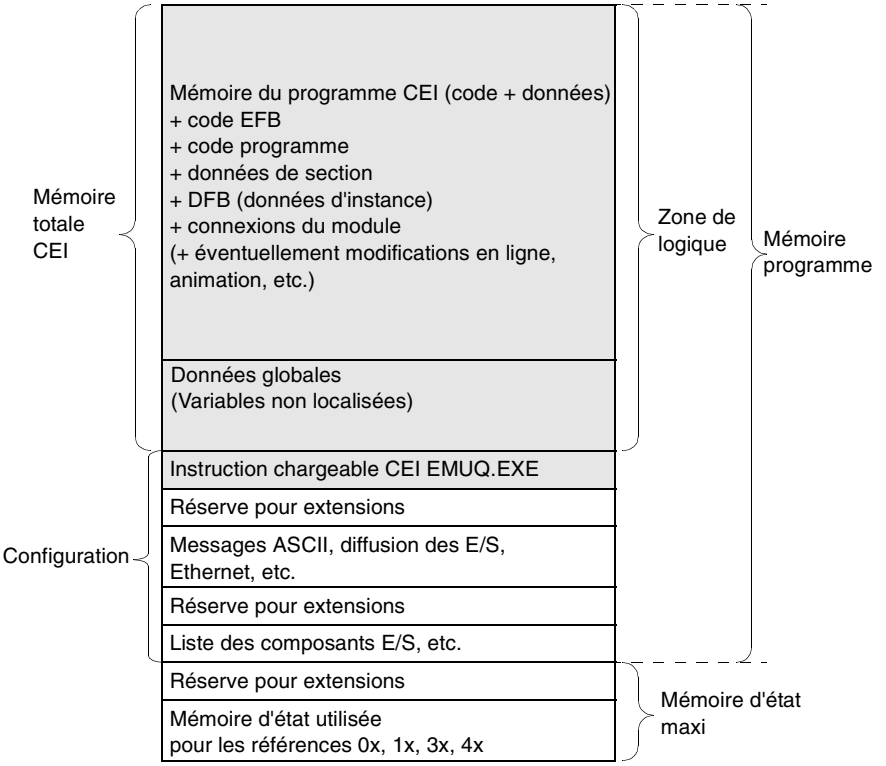
Choisissez donc les langages que vous désirez utiliser :

- *Utilisation exclusive de CEI, p. 163*
- *Utilisation exclusive de LL984, p. 164*
- *Utilisation conjointe de CEI et LL984, p. 165*

Utilisation exclusive de CEI

Si vous voulez utiliser uniquement CEI, chargez le fichier EXEC "QIECVxxx.bin" (non disponible pour l'UC 424 02). Etant donné que ce fichier EXEC ne comprend pas de système d'exécution, vous devez charger le système d'exécution CEI dans l'automate sous la forme d'une instruction chargeable (EMUQ.exe) (voir également *Instructions chargeables, p. 106*). L'instruction chargeable est chargée dans la zone de la mémoire du programme et y occupe de l'espace mémoire.

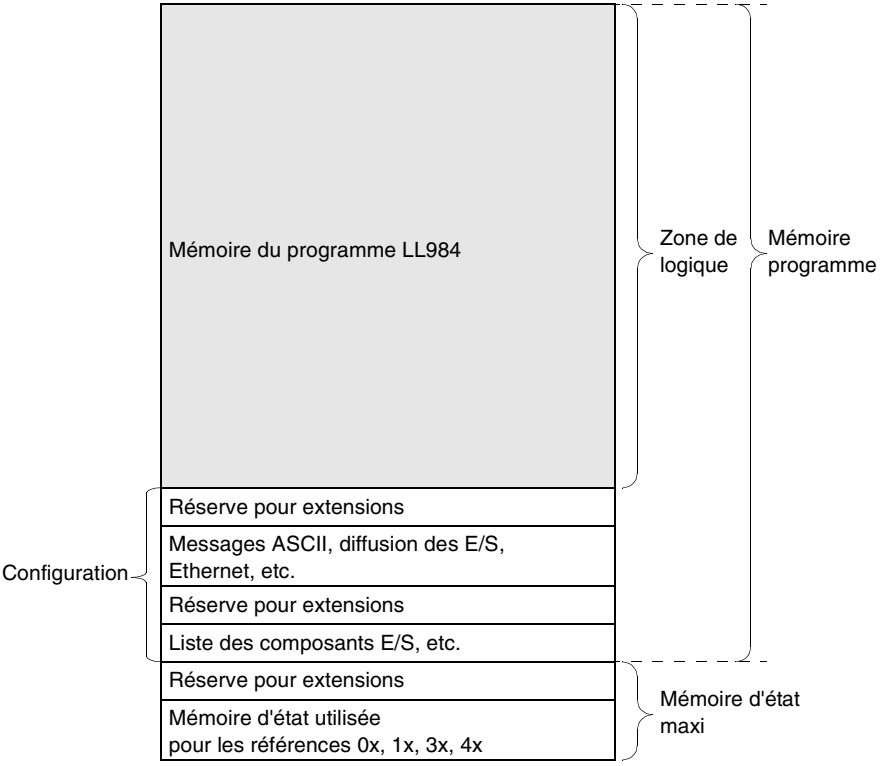
Structure de la mémoire UC X13 0X lors d'une utilisation exclusive de CEI :


Mémoire totale CEI
Mémoire du programme CEI (code + données)
+ code EFB
+ code programme
+ données de section
+ DFB (données d'instance)
+ connexions du module
(+ éventuellement modifications en ligne, animation, etc.)
Zone de logique
Mémoire programme
Données globales
(Variables non localisées)
Configuration
Instruction chargeable CEI EMUQ.EXE
Réserve pour extensions
Messages ASCII, diffusion des E/S, Ethernet, etc.
Réserve pour extensions
Liste des composants E/S, etc.
Réserve pour extensions
Mémoire d'état utilisée pour les références 0x, 1x, 3x, 4x
Mémoire d'état maxi

Utilisation exclusive de LL984

Si vous voulez utiliser uniquement LL984, chargez le fichier EXEC "Q186Vxxx.bin" pour l'UC X13 0X et le fichier EXEC "Q486Vxxx.bin" pour l'UC 424 02.

Structure de la mémoire de l'UC X13 0X lors d'une utilisation exclusive de LL984 :

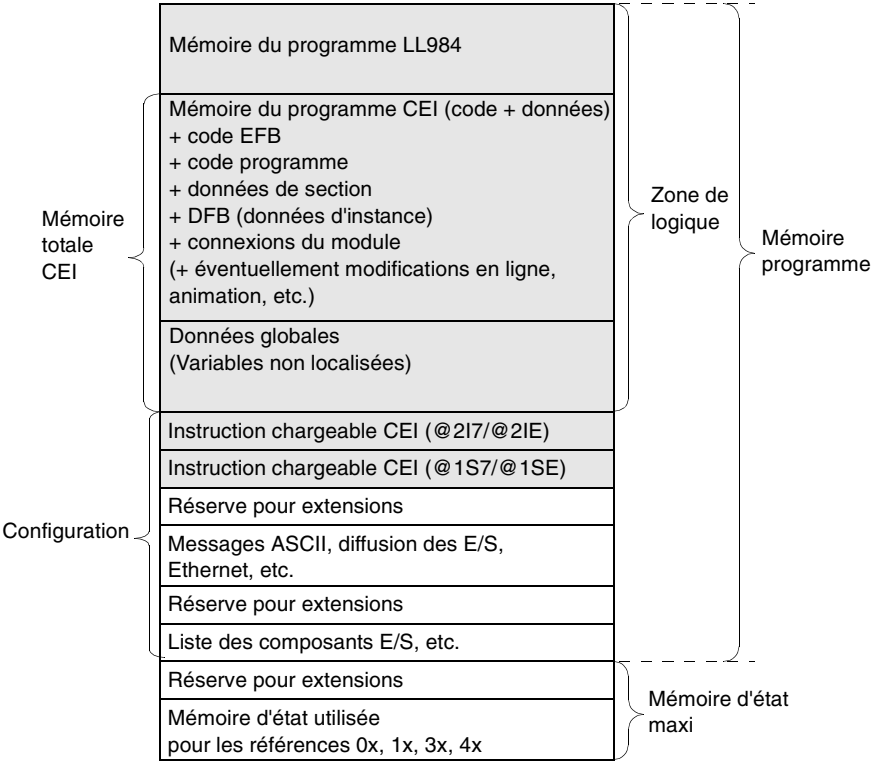

Mémoire du programme LL984
Zone de logique
Mémoire programme
Configuration
Réserve pour extensions
Messages ASCII, diffusion des E/S, Ethernet, etc.
Réserve pour extensions
Liste des composants E/S, etc.
Réserve pour extensions
Mémoire d'état utilisée pour les références 0x, 1x, 3x, 4x
Mémoire d'état maxi

Utilisation conjointe de CEI et LL984

Si vous voulez utiliser conjointement CEI et LL984, chargez le fichier EXEC "Q186Vxxx.bin" pour l'UC X13 0X et le fichier EXEC "Q486Vxxx.bin" pour l'UC 424 02. Etant donné que ces fichiers EXEC ne contiennent que le système d'exécution LL984, vous devez charger dans l'automate le système d'exécution CEI sous la forme d'instructions chargeables (@2I7/@2IE et @1S7/@1SE) (voir également*Instructions chargeables, p. 106*). Ces deux instructions chargeables sont chargées dans la zone de la mémoire du programme et y occupent de l'espace mémoire.

Note : L'utilisation conjointe de CEI et LL984 n'est pas possible avec l'UC 113 02 car sa mémoire est trop petite.

Structure de la mémoire de l'UC X13 0X lors d'une utilisation conjointe de CEI et LL984 :


Mémoire du programme LL984
Mémoire du programme CEI (code + données)
+ code EFB
+ code programme
+ données de section
+ DFB (données d'instance)
+ connexions du module
(+ éventuellement modifications en ligne, animation, etc.)
Données globales
(Variables non localisées)
Instruction chargeable CEI (@2I7/@2IE)
Instruction chargeable CEI (@1S7/@1SE)
Réserve pour extensions
Messages ASCII, diffusion des E/S, Ethernet, etc.
Réserve pour extensions
Liste des composants E/S, etc.
Réserve pour extensions
Mémoire d'état utilisée pour les références 0x, 1x, 3x, 4x
Mémoire totale CEI
Configuration
Zone de logique
Mémoire programme
Mémoire d'état maxi

Utiliser la mémoire étendue (mémoire d'état pour les références 6x)

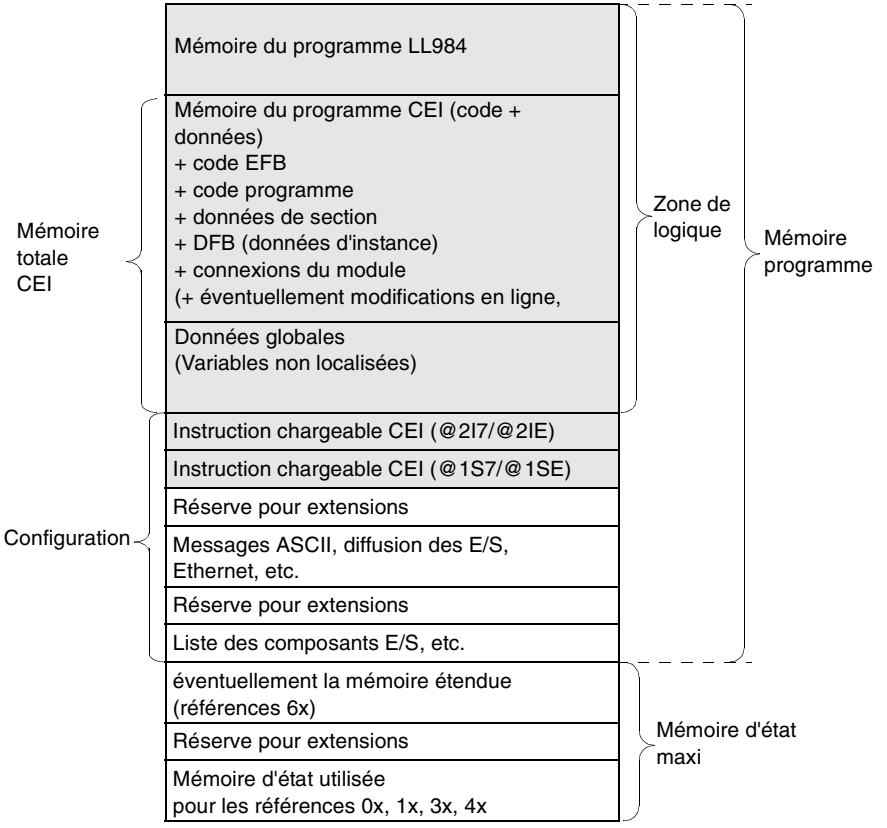
Introduction

Si vous utilisez une UC 213 04 ou UC 424 02, vous pouvez mettre à disposition des références 6x une zone de la mémoire d'état.

Note : Les références 6x sont des mots indicateurs et peuvent uniquement être utilisées avec les programmes utilisateur LL984.

Même lorsque la zone de la mémoire d'état est en dehors de la zone de la mémoire du programme, la taille de la mémoire d'état influence la taille de la mémoire du programme.

Utiliser la mémoire étendue (mémoire d'état pour les références 6x) :


Mémoire du programme LL984
Mémoire du programme CEI (code + données)
+ code EFB
+ code programme
+ données de section
+ DFB (données d'instance)
+ connexions du module
(+ éventuellement modifications en ligne,
Données globales
(Variables non localisées)
Instruction chargeable CEI (@2I7/@2IE)
Instruction chargeable CEI (@1S7/@1SE)
Réserve pour extensions
Messages ASCII, diffusion des E/S, Ethernet, etc.
Réserve pour extensions
Liste des composants E/S, etc.
éventuellement la mémoire étendue (références 6x)
Réserve pour extensions
Mémoire d'état utilisée pour les références 0x, 1x, 3x, 4x
Mémoire totale CEI
Configuration
Zone de logique
Mémoire programme
Mémoire d'état maxi

Si vous n'utilisez PAS 6x

Si vous ne voulez pas utiliser de références 6x, vous pouvez choisir, pour une UC 213 04, si vous réservez de la mémoire d'état pour les références 6x ou non.

Sélectionnez **Logique 48 K/ Mémoire 32 K** dans la zone de liste **Taille mémoire** sous **Projet** → **Configuration automate** → **Sélection automate.**

Note : Pour une UC 424 02, il est impossible de désactiver la plage 6x.

Si vous utilisez 6x

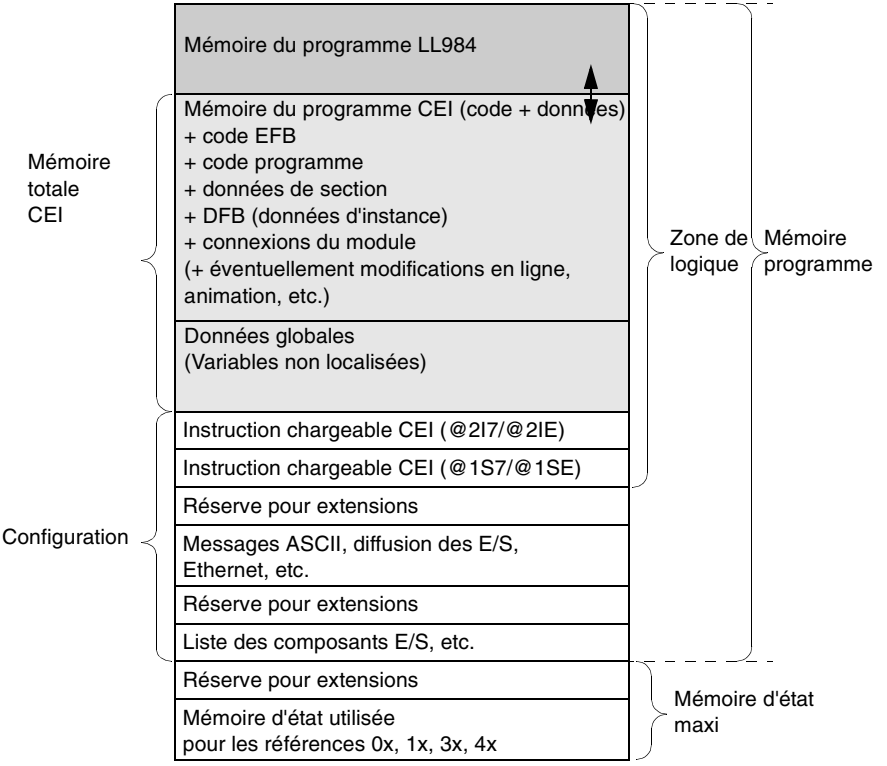
Si vous voulez utiliser les références 6x, sélectionnez **Logique 32 K / Mémoire 64 K** dans la zone de liste **Taille mémoire** sous **Projet** → **Configuration automate** → **Sélection automate**.

Synchronisation de la zone CEI et de la zone LL984

Introduction

Lors d'une utilisation conjointe des sections CEI et LL984 vous devez syntoniser la taille des deux zones l'une par rapport à l'autre.

Synchronisation de la zone CEI et LL984 :


Mémoire du programme LL984
Mémoire du programme CEI (code + données)
+ code EFB
+ code programme
+ données de section
+ DFB (données d'instance)
+ connexions du module
(+ éventuellement modifications en ligne, animation, etc.)
Données globales
(Variables non localisées)
Instruction chargeable CEI (@2I7/@2IE)
Instruction chargeable CEI (@1S7/@1SE)
Réserve pour extensions
Messages ASCII, diffusion des E/S, Ethernet, etc.
Réserve pour extensions
Liste des composants E/S, etc.
Réserve pour extensions
Mémoire d'état utilisée pour les références 0x, 1x, 3x, 4x
Mémoire totale CEI
Configuration
Zone de logique
Mémoire programme
Mémoire d'état maxi

Taille de la zone CEI

Avec la taille de la mémoire de l'instruction chargeable @2I7 et @2IE vous définissez la taille de la mémoire totale CEI et ainsi également l'espace disponible pour les données LL984 (programme utilisateur).

Vous définissez la taille de la mémoire des instructions chargeables dans **Projet** → **Configuration automate** → **Instructions chargeables** → **Installer @2I7 ou @2IE** → **Edition...** → **Taille mémoire**.

La valeur de la taille de la mémoire est indiquée en paragraphes. Un paragraphe représente 16 octets.

Pour les instructions chargeables @1S7 et @1SE aucune taille de mémoire n'est nécessaire. Assurez-vous que "0" est indiqué.

L'espace mémoire total CEI ainsi défini se compose à son tour de différentes zones. La description de la Synchronisation de ces zones est reprise au chapitre *Synchronisation des zones pour les données globales et la mémoire du programme CEI, p. 170*.

Taille de la zone LL984

La taille de la mémoire disponible pour les programmes utilisateur LL984 se calcule avec la formule suivante :

Zone LL984 = nodes LL984 disponibles - taille de la mémoire de l'instruction chargeable @2I7/@2IE - taille de l'instruction chargeable @2I7 et @2IE - taille de l'instruction chargeable @1S7 et @1SE

Lors de ce calcul il convient de prendre en compte le fait que la taille de la zone LL984 se base sur des nodules et que les autres données se basent sur les octets.

Message d'erreur lors du chargement du programme

Si vous obtenez un message d'erreur lors du chargement vous indiquant que le programme utilisateur est trop important pour la mémoire de l'automate, il peut y avoir trois raisons :

1. La mémoire est véritablement trop petite.
2. La taille mémoire de l'instruction chargeable est trop petite (voir le présent chapitre).
3. La zone pour les données globales et la mémoire du programme CEI ne sont pas synchronisées de manière optimale l'une par rapport à l'autre (voir le chapitre *Synchronisation des zones pour les données globales et la mémoire du programme CEI, p. 170*).

Synchronisation des zones pour les données globales et la mémoire du programme CEI

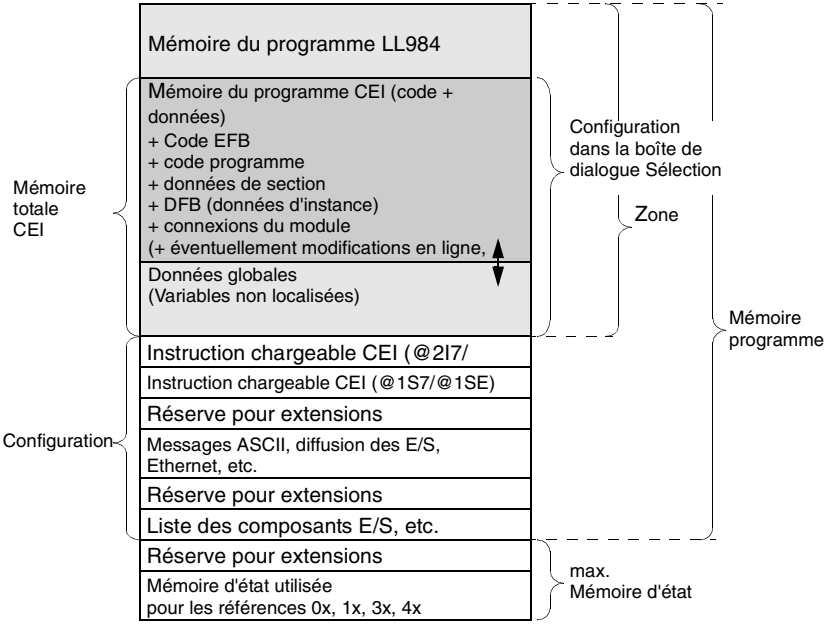
Introduction

L'espace de mémoire totale CEI défini avec la taille de la mémoire de l'instruction chargeable (voir chapitre *Synchronisation de la zone CEI et de la zone LL984, p. 168*) se compose de 2 zones :

- **Mémoire du programme CEI**
  - se composant des codes EFB,
  - des codes du programme,
  - les données de section,
  - les données d'instance DFB,
  - des liaisons de module,
  - éventuellement des données des modifications en ligne,
  - et éventuellement des données d'animation, etc.
- **Données globales**
  - se composant des variables non localisées

Les zones pour les données globales et la mémoire du programme CEI peuvent être synchronisées l'une par rapport à l'autre.

Synchronisation des zones pour les données globales et la mémoire du programme CEI :


Mémoire du programme LL984
Mémoire du programme CEI (code + données)
+ Code EFB
+ code programme
+ données de section
+ DFB (données d'instance)
+ connexions du module
(+ éventuellement modifications en ligne,
Données globales
(Variables non localisées)
Instruction chargeable CEI (@2I7/
Instruction chargeable CEI (@1S7/@1SE)
Réserve pour extensions
Messages ASCII, diffusion des E/S, Ethernet, etc.
Réserve pour extensions
Liste des composants E/S, etc.
Réserve pour extensions
Mémoire d'état utilisée pour les références 0x, 1x, 3x, 4x
Mémoire totale CEI
Configuration
Configuration dans la boîte de dialogue Sélection
Zone
Mémoire programme
max. Mémoire d'état

Taille de la zone de la mémoire programme CEI

Vous pouvez définir la configuration pour la mémoire du programme CEI dans **Projet** → **Configuration automate** → **Sélection automate** dans la zone **CEI**. Indiquez ici la taille de la mémoire totale CEI et des données globales afin d'obtenir la taille de la mémoire du programme CEI (mémoire du programme CEI = mémoire totale CEI - données globales). La configuration n'est possible que si le PC et l'automate sont hors ligne. Si vous n'utilisez pas ou peu de variables non localisées, et faites de même pour les liaisons modules, vous pouvez définir une taille importante de la mémoire programme CEI car vous ne devez pratiquement pas laisser de mémoire pour les données globales.

Taille de la zone pour les données globales

La zone pour les données globales (variables non localisées) se calcule avec la formule suivante :

Zone pour les données globales = taille de la mémoire de l'instruction chargeable - mémoire du programme CEI

L'occupation actuelle des différentes zones (EFB, données d'instance, programme utilisateur, etc.) est affichée sous **En ligne** → **Statistiques mémoire...** → **Statistiques mémoire**. Cet affichage n'est possible que si le PC et l'automate sont en ligne.

Message d'erreur lors du chargement du programme

Si vous obtenez un message d'erreur lors du chargement du programme dans l'automate vous indiquant que le programme utilisateur est trop important pour la mémoire de l'automate, il peut y avoir trois raisons :

1. La mémoire est véritablement trop petite.
2. La taille mémoire de l'instruction chargeable est trop petite (voir le chapitre *Synchronisation de la zone CEI et de la zone LL984, p. 168*).
3. La zone pour les données globales et la zone de mémoire du programme CEI ne sont pas synchronisées de manière optimale l'une par rapport à l'autre (voir le présent chapitre).

6.4 Optimisation de la mémoire pour l'UC 434 12(A) et 534 14(A/B) Quantum

Introduction

Vue d'ensemble

Cette section décrit l'optimisation de la mémoire pour les UC 434 12(A) et 534 14(A/B) Quantum.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Généralités sur l'optimisation de la mémoire pour l'UC 434 12(A) et 534 14(A/B) Quantum	174
Synchronisation de la zone CEI et de la zone LL984	176
Synchronisation des zones pour les données globales et la mémoire du programme CEI (UC 434 12(A) / 534 14(A/B))	181

Généralités sur l'optimisation de la mémoire pour l'UC 434 12(A) et 534 14(A/B) Quantum

Mémoire logique

La zone de la mémoire du programme qui accueille votre programme utilisateur est appelée zone de logique. Cette zone détermine également la taille maximale de votre programme utilisateur.

La taille actuelle de la zone de logique peut être visualisée sous **Projets** → **Configuration automate** dans la Vue d'ensemble de la configuration dans la zone **Automate**. La taille de la mémoire est affichée en nodules pour LL984 (1 nodule équivaut à 11 octets) et en ko pour CEI.

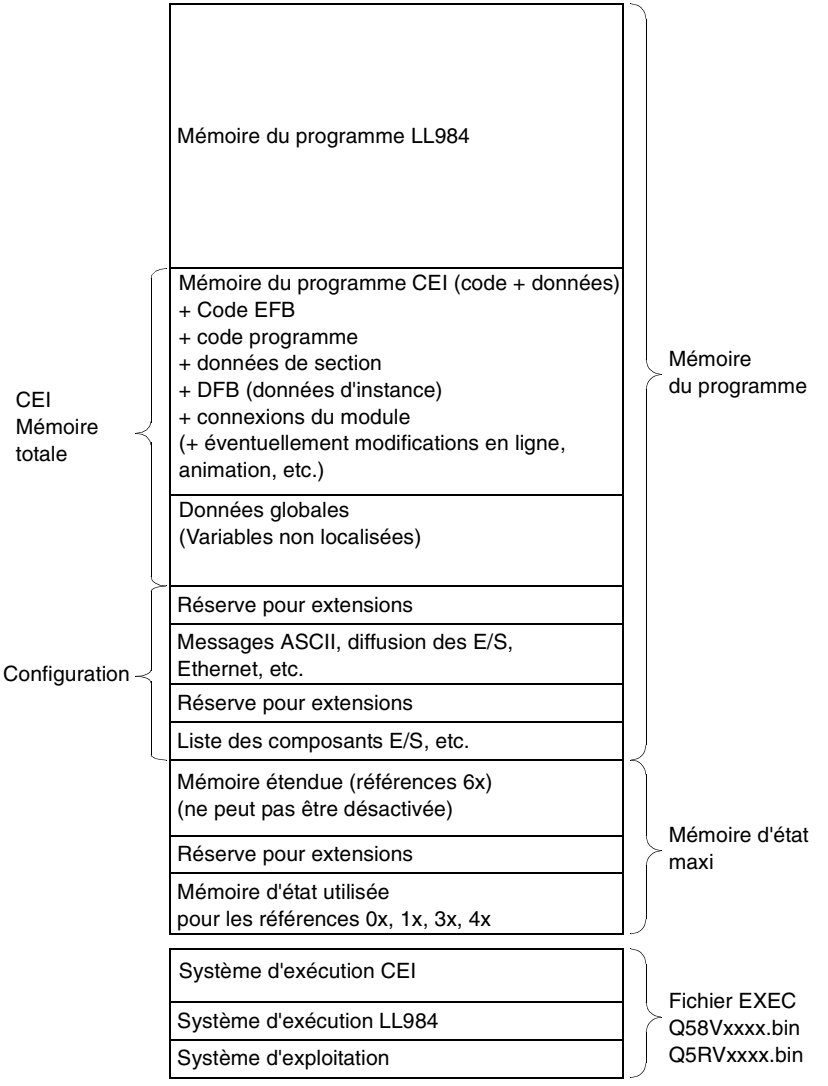
Optimisation de la mémoire logique

Vous avez différentes possibilités pour optimiser la mémoire logique en fonction de vos besoins :

- *Synchronisation de la zone CEI et de la zone LL984, p. 176*
- *Synchronisation des zones pour les données globales et la mémoire du programme CEI (UC 434 12(A) / 534 14(A/B)), p. 181*

Note : Prenez également en compteles possibilités d'optimisation de la mémoire indépendantes de l'automate (voir *Remarques générales sur l'optimisation de la mémoire, p. 153*).

Structure de la mémoire de l'UC 434 12(A) / 534 14(A/B) (représentation simplifiée) :



Synchronisation de la zone CEI et de la zone LL984

Introduction

Pour l'UC 434 12 et 534 14 vous avez besoin du fichier EXEC "Q58Vxxxx.bin".

Pour l'UC 434 12A et 534 14A/B (UC reconçues) vous avez besoin du fichier EXEC "Q5RVxxxx.bin".

Ces fichiers EXEC contiennent les systèmes d'exécution pour CEI et LL984.

La taille des zones logiques pour CEI et LL984 doit être synchronisée l'une par rapport à l'autre. Vous pouvez définir la taille des deux zones dans **Projet** → **Configuration automate** → **Sélection automate**.

En fonction de la taille choisie pour la zone CEI, des zones sont réservées dans la mémoire du programme de l'automate pour CEI et/ou LL984. Si vous définissez une zone combinée CEI et LL984 et qu'ensuite vous n'utilisez qu'un seul des deux types de langage dans votre programme utilisateur, la mémoire du programme n'est pas utilisée de manière optimale.

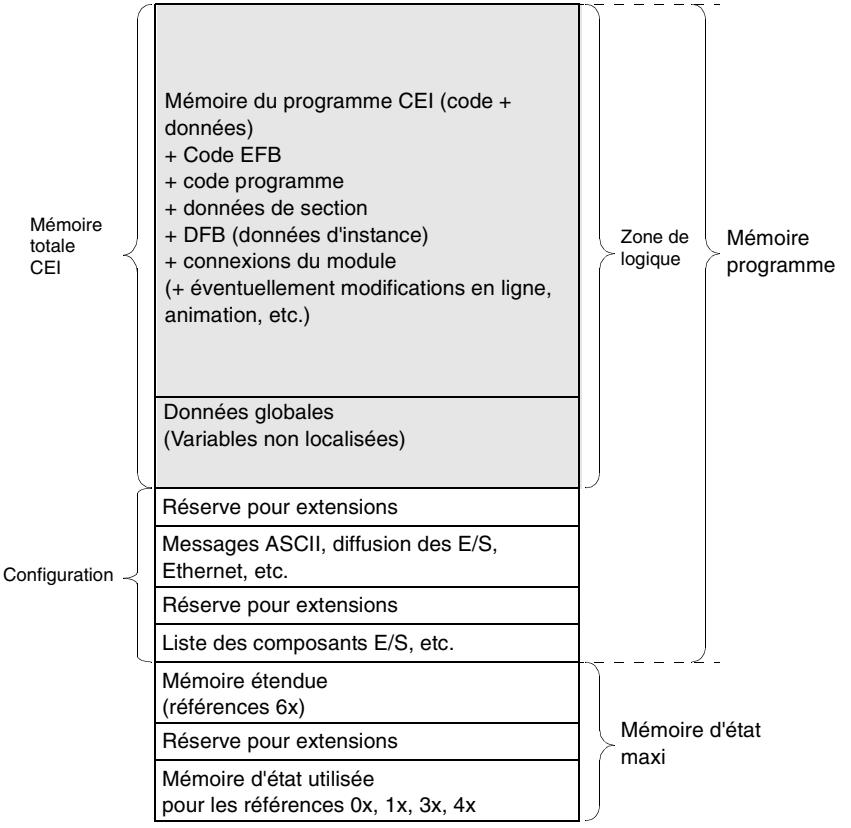
Choisissez donc les langages que vous désirez utiliser :

- *Utilisation exclusive de CEI, p. 177*
- *Utilisation exclusive de LL984, p. 178*
- *Utilisation conjointe de CEI et LL984, p. 179*

Utilisation exclusive de CEI

Si vous utilisez uniquement le langage CEI, sélectionnez l'entrée **Validée** sous **Projet** → **Configuration automate** → **Sélection automate** dans la zone de liste **Moteur d'exécution** et déplacez le curseur **Taille du tas CEI** jusqu'à la bordure droite (valeur maximale). Ainsi, la zone LL984 est entièrement désactivée et vous disposez alors de toute la zone totale de logique pour votre programme utilisateur CEI.

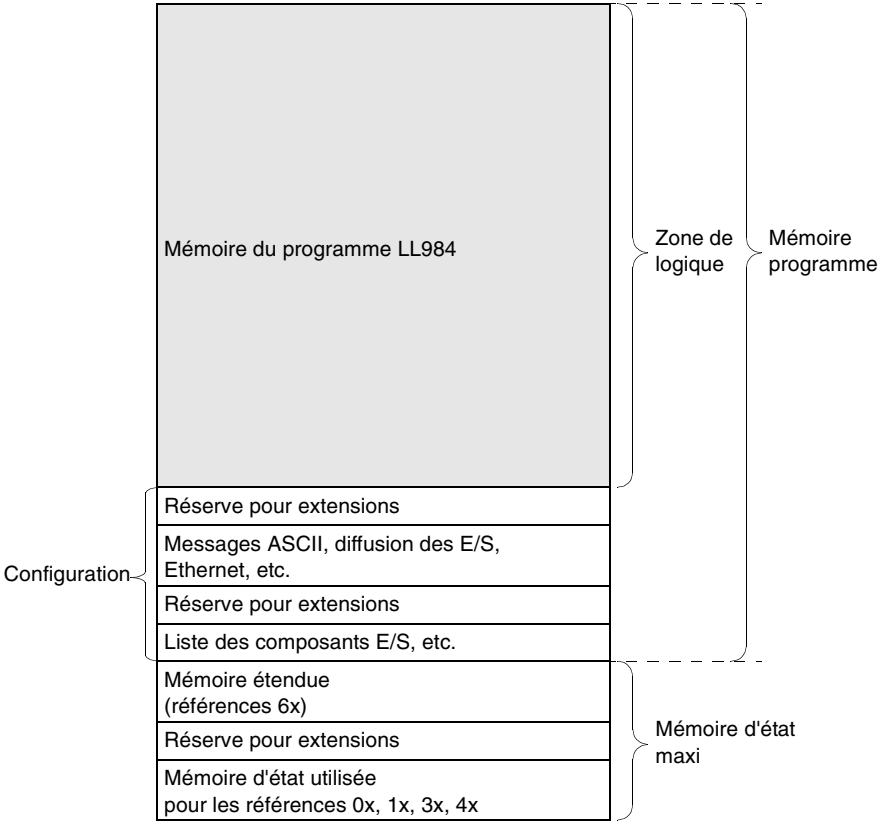
Structure de la mémoire de l'UC 434 12(A) / 534 14(A/B) lors de l'utilisation exclusive de CEI :



Utilisation exclusive de LL984

Vous utilisez uniquement le langage LL984, sélectionnez l'entrée **Invalider** sous **Projet** → **Configuration automate** → **Sélection automate** dans la zone de liste **Moteur d'exécution**. Ainsi, la zone CEI est entièrement désactivée et vous disposez alors de toute la zone totale de logique pour votre programme utilisateur LL984.

Structure de la mémoire de l'UC 434 12(A) / 534 14(A/B) lors de l'utilisation exclusive de LL984 :


Mémoire du programme LL984
Zone de logique
Mémoire programme
Configuration
Réserve pour extensions
Messages ASCII, diffusion des E/S, Ethernet, etc.
Réserve pour extensions
Liste des composants E/S, etc.
Mémoire étendue (références 6x)
Réserve pour extensions
Mémoire d'état utilisée pour les références 0x, 1x, 3x, 4x
Mémoire d'état maxi

Utilisation conjointe de CEI et LL984

Lors d'une utilisation conjointe de CEI et LL984 vous devez syntoniser la taille des deux zones.

En configurant l'élément **Taille du tas CEI** et **Données globales**, vous définissez automatiquement la taille de la mémoire du programme CEI et par là-même définissez la place disponible pour les données LL984 (programme utilisateur).

La taille de la mémoire disponible pour les programmes utilisateur LL984 se calcule avec la formule suivante :

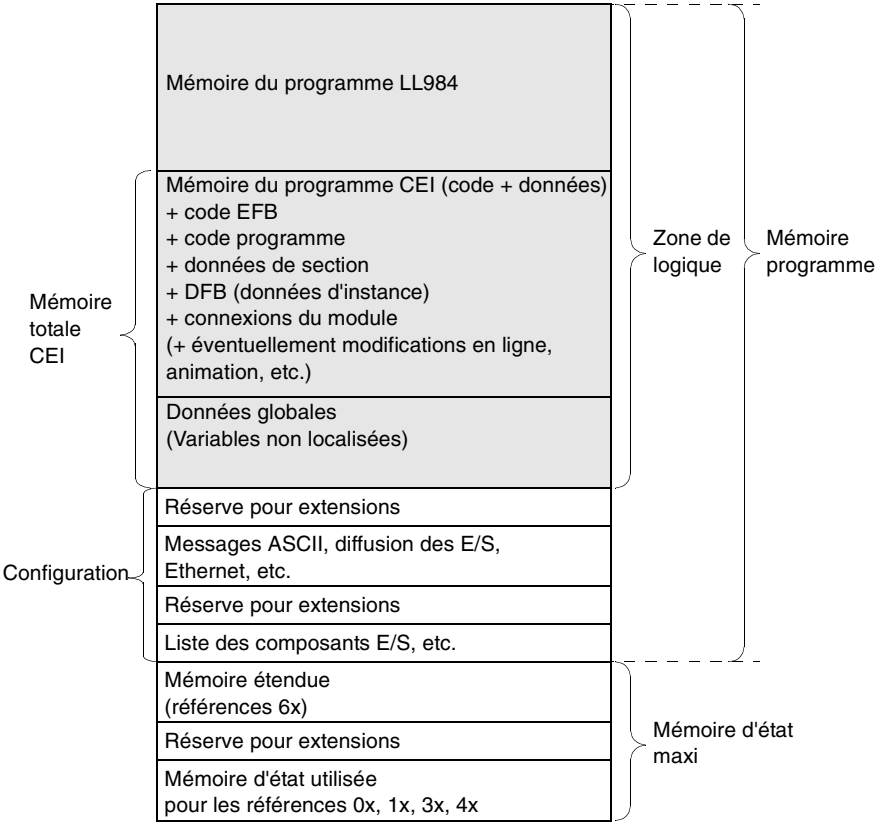
Zone LL984 = nodules LL984 disponibles - mémoire totale CEI

Lors de ce calcul il convient de prendre en compte le fait que la taille de la zone LL984 est calculée en nodules et que les autres données sont calculées en ko.

Pour configurer la mémoire totale CEI, sélectionnez l'entrée **Validée** dans la zone de liste **Moteur d'exécution** sous **Projet** → **Configuration automate** → **Sélection automate**. La zone CEI est alors validée et vous pouvez saisir la taille de mémoire voulue dans la zone de texte **Taille du tas CEI**. La taille de la mémoire est indiquée en ko.

L'espace mémoire total CEI ainsi défini se compose à son tour de différentes zones. La description de la synchronisation de ces zones est reprise au chapitre *Synchronisation des zones pour les données globales et la mémoire du programme CEI, p. 170*.

Structure de la mémoire de l'UC 434 12(A) / 534 14(A/B) lors de l'utilisation conjointe de CEI et LL984 :



Message d'erreur lors du chargement du programme

Si vous obtenez un message d'erreur lors du chargement vous indiquant que le programme utilisateur est trop important pour la mémoire de l'automate, il peut y avoir trois raisons :

1. La mémoire est véritablement trop petite.
2. La zone de logique est trop petite (voir le présent chapitre).
3. La zone pour les données globales et pour la mémoire du programme CEI ne sont pas synchronisées de manière optimale l'une par rapport à l'autre (voir le chapitre *Synchronisation des zones pour les données globales et la mémoire du programme CEI (UC 434 12(A) / 534 14(A/B)), p. 181*).

Synchronisation des zones pour les données globales et la mémoire du programme CEI (UC 434 12(A) / 534 14(A/B))

Introduction

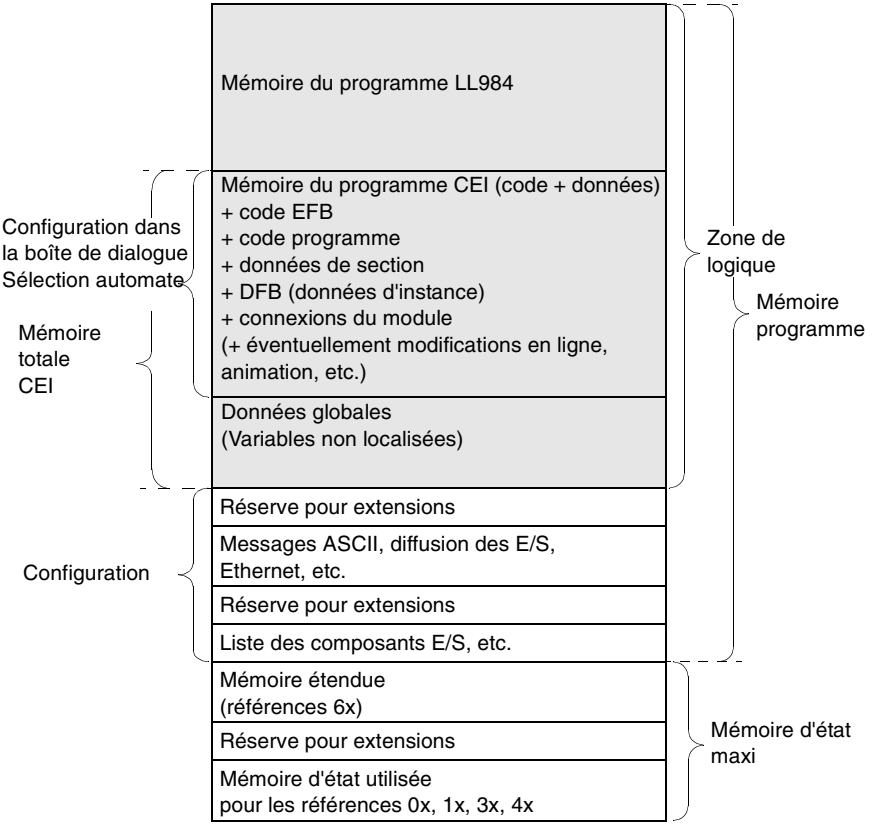
La mémoire totale CEI définie (voir chapitre *Synchronisation de la zone CEI et de la zone LL984, p. 176*) se compose de 2 zones.

L'espace de mémoire totale CEI défini avec la taille de la mémoire de l'instruction chargeable (voir chapitre *Synchronisation de la zone CEI et de la zone LL984, p. 168*) se compose de 2 zones :

- **Mémoire du programme CEI**
  - se composant des codes EFB,
  - des codes programme,
  - des données de section,
  - des données d'instance DFB,
  - des liaisons de module,
  - éventuellement des données des modifications en ligne,
  - éventuellement des données d'animation, etc.
- **Données globales**
  - se composant des variables non localisées

Les zones pour les données globales et la mémoire du programme CEI peuvent être synchronisées l'une par rapport à l'autre.

Synchronisation des zones pour les données globales et la mémoire du programme CEI (UC 434 12(A) / 534 14(A/B))



Taille de la zone de la mémoire du programme CEI

Vous pouvez définir la configuration pour la mémoire du programme CEI dans **Projet** → **Configuration automate** → **Sélection automate** dans la zone **CEI**. Indiquez ici la taille de la mémoire totale CEI et des données globales afin d'obtenir la taille de la mémoire du programme CEI (mémoire du programme CEI = mémoire totale CEI - données globales). La configuration n'est possible que si le PC et l'automate sont hors ligne. Si vous n'utilisez pas ou peu de variables non localisées, et faites de même pour les liaisons modules, vous pouvez définir une taille importante de la mémoire programme CEI car vous ne devez pratiquement pas laisser de mémoire pour les données globales.

Taille de la zone pour les données globales

La zone pour les données globales (variables non localisées) se calcule avec la formule suivante :

Zone pour les données globales = taille de la mémoire de l'instruction chargeable - mémoire du programme CEI

L'occupation actuelle des différentes zones (EFB, données d'instance, programme utilisateur, etc.) est affichée sous **En ligne** → **Statistiques mémoire...** → **Statistiques mémoire**. Cet affichage n'est possible que si le PC et l'automate sont en ligne.

Message d'erreur lors du chargement du programme

Si vous obtenez un message d'erreur lors du chargement du programme dans l'automate vous indiquant que le programme utilisateur est trop important pour la mémoire de l'automate, il peut y avoir trois raisons :

1. La mémoire est véritablement trop petite.
2. La taille de la mémoire totale CEI est trop petite (voir le chapitre *Synchronisation de la zone CEI et de la zone LL984, p. 176*).
3. La zone pour les données globales et la zone de mémoire du programme CEI ne sont pas synchronisées de manière optimale l'une par rapport à l'autre (voir le présent chapitre).

6.5 Optimisation de mémoire pour les UC Compact

Introduction

Aperçu

Ce chapitre décrit l'optimisation de mémoire pour les UC Compact.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Généralités sur l'optimisation de la mémoire pour les UC Compact	186
Synchronisation de la zone CEI et de la zone LL984	188
Synchronisation des zones pour les données globales et la mémoire du programme CEI (Compact)	193

Généralités sur l'optimisation de la mémoire pour les UC Compact

Mémoire logique

La zone de la mémoire du programme qui accueille votre programme utilisateur est appelée zone de logique. Cette zone détermine également la taille maximale de votre programme utilisateur.

La taille actuelle de la zone de logique peut être visualisée sous **Projets** → **Configuration automate** dans la Vue d'ensemble de la configuration dans la zone **Automate**. La taille de la mémoire est affichée en nodules pour LL984 (1 nodule équivaut à 11 octets) et en ko pour CEI.

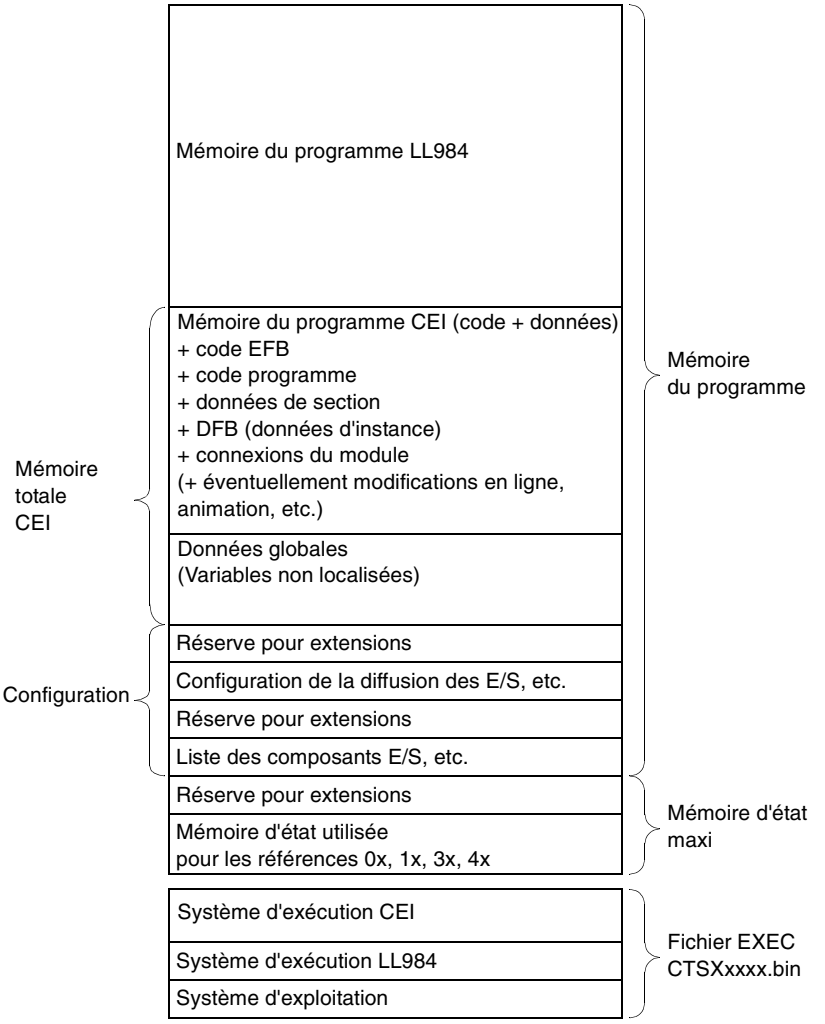
Optimisation de la mémoire logique

Vous avez différentes possibilités pour optimiser la mémoire logique en fonction de vos besoins :

- *Synchronisation de la zone CEI et de la zone LL984, p. 188*
- *Synchronisation des zones pour les données globales et la mémoire du programme CEI (Compact), p. 193*

Note : Prenez également en compteles possibilités d'optimisation de la mémoire indépendantes de l'automate (voir *Remarques générales sur l'optimisation de la mémoire, p. 153*).

Structure de la mémoire d'une UC Compact (représentation simplifiée) :



Synchronisation de la zone CEI et de la zone LL984

Introduction

Le fichier EXEC nécessaire pour les UC Compact "CTSXxxxx.bin" contient les systèmes d'exploitation pour CEI et LL984 (voir également *Introduction à l'installation*).

La taille des zones logiques pour CEI et LL984 doit être synchronisée l'une par rapport à l'autre. Vous pouvez définir la taille des deux zones dans **Projet** → **Configurateur automate** → **Sélection automate**.

En fonction de la taille choisie pour la zone CEI, des zones sont réservées dans la mémoire du programme de l'automate pour CEI et/ou LL984. Si vous définissez une zone combinée CEI et LL984 et qu'ensuite vous n'utilisez qu'un seul des deux types de langage dans votre programme utilisateur, la mémoire du programme n'est pas utilisée de manière optimale.

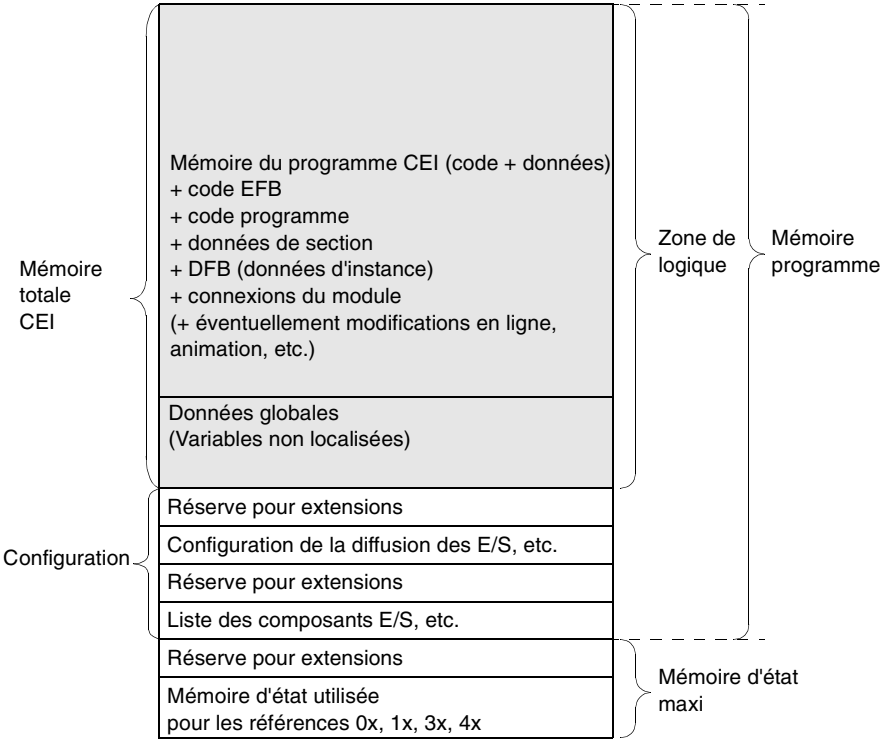
Choisissez donc les langages que vous désirez utiliser :

- *Utilisation exclusive de CEI, p. 189*
- *Utilisation exclusive de LL984, p. 190*
- *Utilisation conjointe de CEI et LL984, p. 191*

Utilisation exclusive de CEI

Si vous utilisez uniquement le langage CEI, sélectionnez l'entrée **Validée** sous **Projet** → **Configuration automate** → **Sélection automate** dans la zone de liste **Moteur d'exécution CEI** et déplacez le curseur **Taille du tas CEI** jusqu'à la bordure droite (valeur maximale).Ainsi, la zone LL984 est entièrement désactivée et vous disposez alors de toute la zone de logique pour votre programme utilisateur CEI.

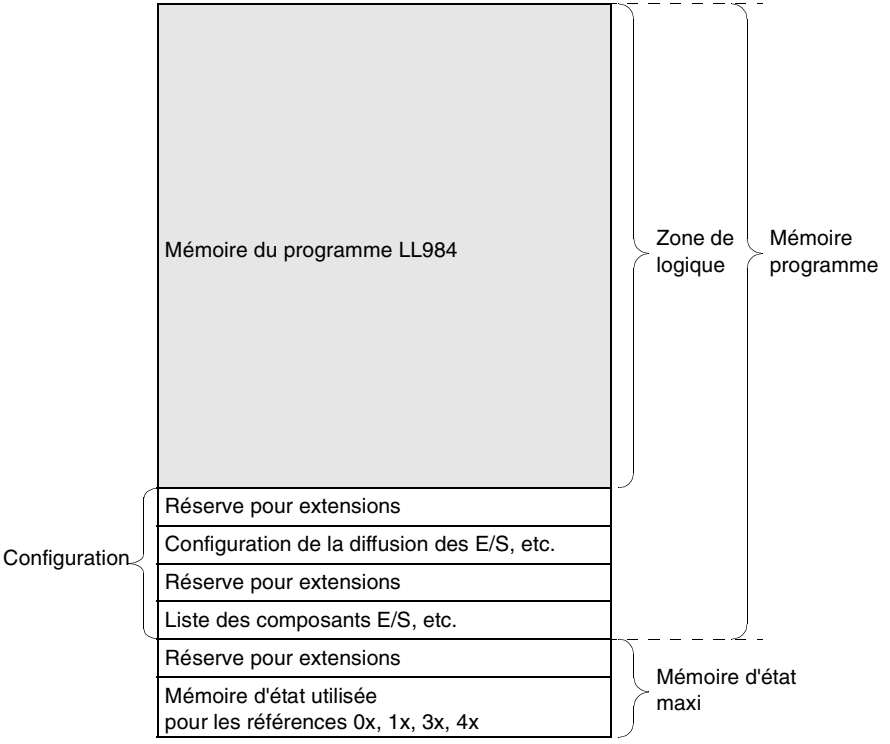
Structure de la mémoire des UC Compact lors d'une utilisation exclusive de CEI :



Utilisation exclusive de LL984

Si vous utilisez uniquement le langage LL984, sélectionnez l'entrée **Invalider** sous **Projet** → **Configuration automate** → **Sélection automate** dans la zone de liste **Moteur d'exécution CEI**. Ainsi, la zone CEI est entièrement désactivée et vous disposez alors de toute la zone de logique pour votre programme utilisateur LL984.

Structure de la mémoire des UC Compact lors d'une utilisation exclusive de LL984 :


Mémoire du programme LL984
Zone de logique
Mémoire programme
Configuration
Réserve pour extensions
Configuration de la diffusion des E/S, etc.
Réserve pour extensions
Liste des composants E/S, etc.
Réserve pour extensions
Mémoire d'état utilisée pour les références 0x, 1x, 3x, 4x
Mémoire d'état maxi

Utilisation conjointe de CEI et LL984

Lors d'une utilisation conjointe de CEI et LL984 vous devez syntoniser la taille des deux zones.

En configurant l'élément **Taille du tas CEI** et **Données globales**, vous définissez automatiquement la taille de la mémoire du programme CEI et par là-même définissez la place disponible pour les données LL984 (programme utilisateur).

La taille de la mémoire disponible pour les programmes utilisateur LL984 se calcule avec la formule suivante :

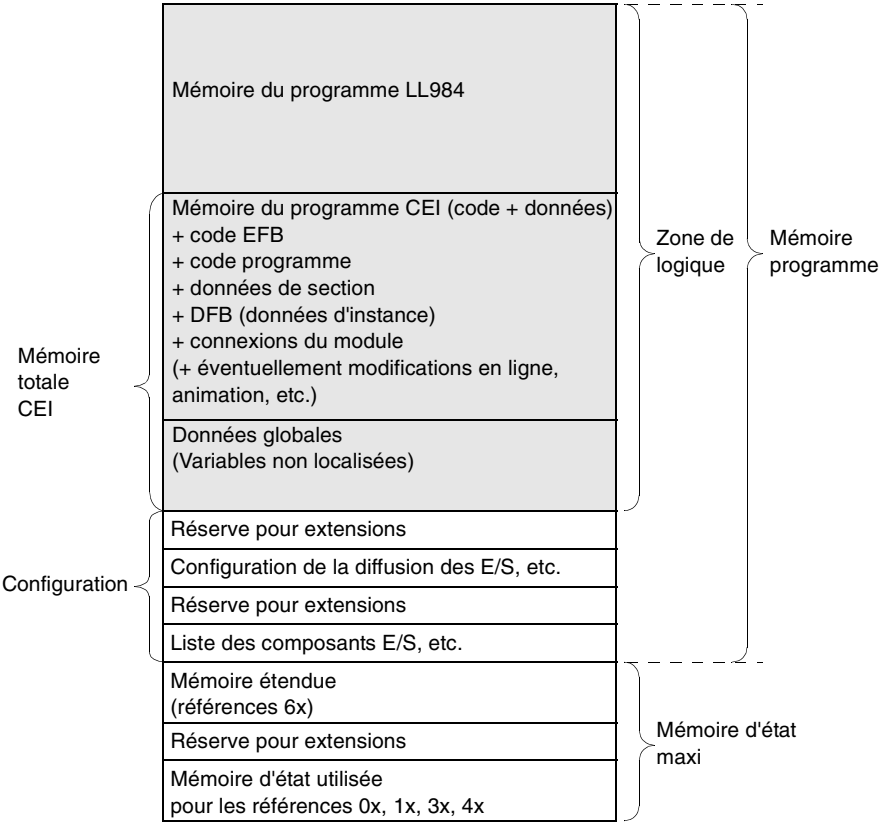
Zone LL984 = nodules LL984 disponibles - mémoire totale CEI

Lors de ce calcul il convient de prendre en compte le fait que la taille de la zone LL984 est calculée en nodules et que les autres données sont calculées en ko.

Pour configurer la mémoire totale CEI, sélectionnez l'entrée **Validée** dans la zone de liste **Moteur d'exécution** sous **Projet** → **Configuration automate** → **Sélection automate**. La zone CEI est alors validée et vous pouvez saisir la taille de mémoire voulue dans la zone de texte **Taille du tas CEI**. La taille de la mémoire est indiquée en ko.

L'espace mémoire total CEI ainsi défini se compose à son tour de différentes zones. La description de la synchronisation de ces zones est reprise au chapitre *Synchronisation des zones pour les données globales et la mémoire du programme CEI (Compact), p. 193*.

Structure de la mémoire Compact lors d'une utilisation conjointe de CEI et LL984 :



Message d'erreur lors du chargement du programme

Si vous obtenez un message d'erreur lors du chargement vous indiquant que le programme utilisateur est trop important pour la mémoire de l'automate, il peut y avoir trois raisons :

1. La mémoire est véritablement trop petite.
2. La zone de logique est trop petite (voir le présent chapitre).
3. La zone pour les données globales et la mémoire du programme CEI ne sont pas synchronisées de manière optimale l'une par rapport à l'autre (voir le chapitre *Synchronisation des zones pour les données globales et la mémoire du programme CEI (Compact), p. 193*).

Synchronisation des zones pour les données globales et la mémoire du programme CEI (Compact)

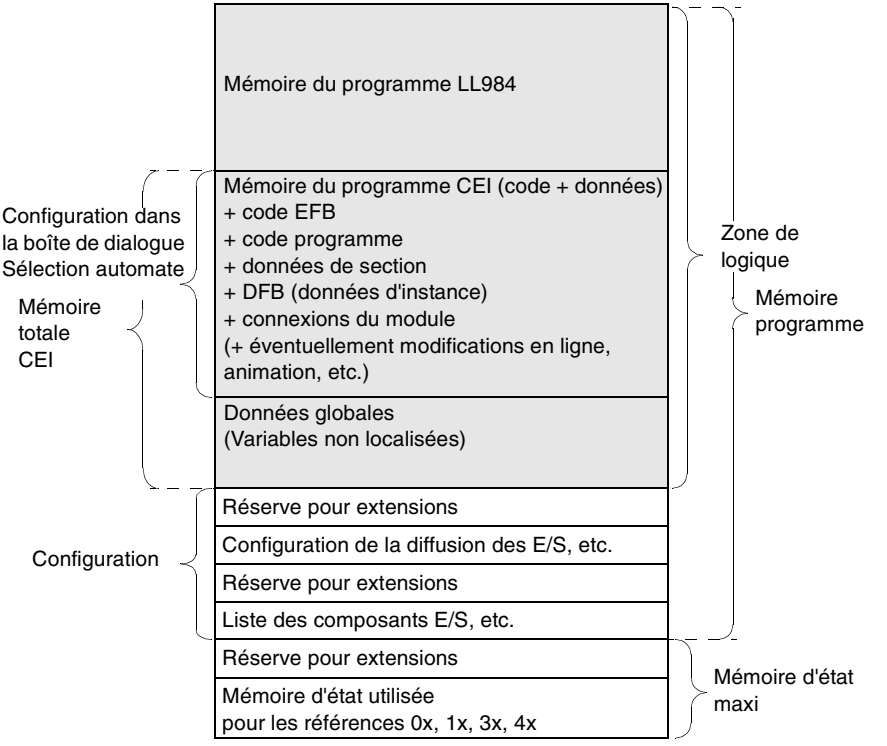
Introduction

La mémoire totale CEI définie (voir chapitre *Synchronisation de la zone CEI et de la zone LL984, p. 188*) se compose de 2 zones.

- **Mémoire du programme CEI**
  - se composant des codes EFB,
  - des codes programme,
  - des données de section,
  - des données d'instance DFB,
  - des liaisons de module,
  - éventuellement des données des modifications en ligne,
  - éventuellement des données d'animation, etc.
- **Données globales**
  - se composant des variables non localisées

Les zones pour les données globales et la mémoire du programme CEI peuvent être synchronisées l'une par rapport à l'autre.

Synchronisation des zones pour les données globales et la mémoire du programme CEI (Compact) :


Mémoire du programme LL984
Mémoire du programme CEI (code + données)
+ code EFB
+ code programme
+ données de section
+ DFB (données d'instance)
+ connexions du module
(+ éventuellement modifications en ligne, animation, etc.)
Données globales
(Variables non localisées)
Réserve pour extensions
Configuration de la diffusion des E/S, etc.
Réserve pour extensions
Liste des composants E/S, etc.
Réserve pour extensions
Mémoire d'état utilisée
pour les références 0x, 1x, 3x, 4x
Configuration dans la boîte de dialogue Sélection automate
Mémoire totale CEI
Configuration
Zone de logique
Mémoire programme
Mémoire d'état maxi

Taille de la zone de la mémoire du programme CEI

La configuration de la Mémoire du programme CEI se fait sous **Projet** → **Configuration automate** → **Sélection automate** dans la zone **CEI**. Indiquez ici la taille de la mémoire totale CEI et des données globales afin d'obtenir la taille de la mémoire du programme CEI (mémoire du programme CEI = mémoire totale CEI - données globales). La configuration n'est possible que si le PC et l'automate sont hors ligne. Si vous n'utilisez pas ou peu de variables non localisées, et faites de même pour les liaisons modules, vous pouvez définir une taille importante de la mémoire programme CEI car vous ne devez pratiquement pas laisser de mémoire pour les données globales.

Taille de la zone pour les données globales

La zone pour les données globales (variables non localisées) se calcule avec la formule suivante :

Zone pour les données globales = taille de la mémoire de l'instruction chargeable - mémoire du programme CEI

L'occupation actuelle des différentes zones (EFB, données d'instance, programme utilisateur, etc.) est affichée sous **En ligne** → **Statistiques mémoire...** → **Statistiques mémoire**. Cet affichage n'est possible que si le PC et l'automate sont en ligne.

Message d'erreur lors du chargement du programme

Si vous obtenez un message d'erreur lors du chargement du programme dans l'automate vous indiquant que le programme utilisateur est trop important pour la mémoire de l'automate, il peut y avoir trois raisons :

1. La mémoire est véritablement trop petite.
2. La taille de la mémoire totale CEI est trop petite (voir le chapitre *Synchronisation de la zone CEI et de la zone LL984, p. 188*).
3. La zone pour les données globales et la zone de mémoire du programme CEI ne sont pas synchronisées de manière optimale l'une par rapport à l'autre (voir le présent chapitre).

6.6 Optimisation de mémoire pour les UC Momentum

Introduction

Vue d'ensemble

Ce chapitre décrit l'optimisation de mémoire pour les UC Momentum.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Généralités sur l'optimisation de la mémoire pour les UC Momentum	198
Sélectionner le fichier EXEC optimal	200
Synchronisation des zones pour les données globales et la mémoire du programme CEI (Momentum)	201

Généralités sur l'optimisation de la mémoire pour les UC Momentum

Mémoire logique

La zone de la mémoire du programme qui accueille votre programme utilisateur est appelée zone de logique. Cette zone détermine également la taille maximale de votre programme utilisateur.

La taille actuelle de la zone de logique peut être visualisée sous **Projet** → **Configuration automate** dans la vue d'ensemble de la configuration dans la zone **Automate**. La taille de la mémoire est affichée en nodules pour LL984 (1 nodule équivaut à 11 octets) et en ko pour CEI.

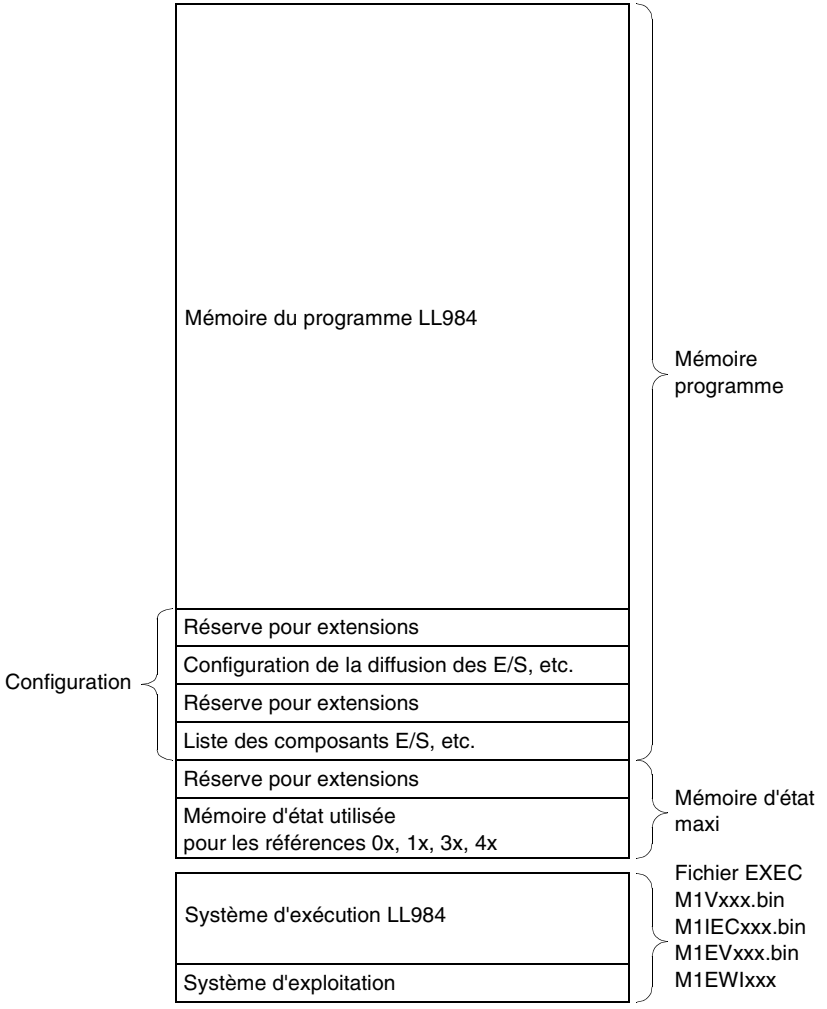
Optimisation de la mémoire logique

Vous avez différentes possibilités pour optimiser la mémoire logique en fonction de vos besoins :

- *Sélectionner le fichier EXEC optimal, p. 200*
- *Synchronisation des zones pour les données globales et la mémoire du programme CEI (Momentum), p. 201*

Note : Prenez également en compteles possibilités d'optimisation de la mémoire indépendantes de l'automate (voir *Remarques générales sur l'optimisation de la mémoire, p. 153*).

Structure de la mémoire d'une UC Momentum (représentation simplifiée) :



Sélectionner le fichier EXEC optimal

Introduction

Avec Momentum, il n'est pas possible d'utiliser ensemble CEI et LL984.

Utilisation de CEI

Affectation des fichiers EXEC lors de l'utilisation de CEI :

171 CBB	M1IVxxxE	MPSV100e.BIN
970 30	-	x

171 CCS	M1IVxxxE	M1EVxxxE
760 00	x	-
760 10	x	-
780 10	x	-
960 30	-	x
980 30	-	x

Utilisation de LL984

Affectation des fichiers EXEC lors de l'utilisation de LL984 :

171 CBB	M1LLVxxx	M1MVxxxE
970 30	x	-

171 CCS	M1LLVxxx	M1EVxxx
700 10	x	-
700/780 00	x	-
760 00	x	-
760 10	x	-
780 10	x	-
960 20	-	x
960 30	-	x
980 20	-	x
980 30	-	x

Synchronisation des zones pour les données globales et la mémoire du programme CEI (Momentum)

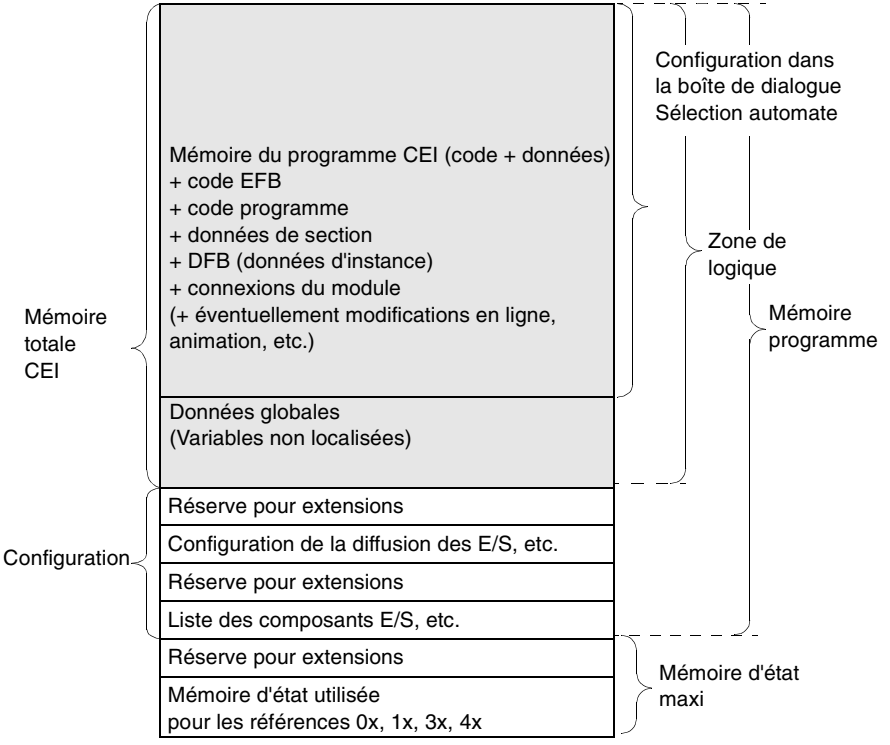
Introduction

La zone de logique pour la mémoire totale CEI se compose de 2 zones.

- **Mémoire du programme CEI**
  - se composant des codes EFB,
  - des codes programme,
  - des données de section,
  - des données d'instance DFB,
  - des liaisons de module,
  - éventuellement des données des modifications en ligne,
  - éventuellement des données d'animation, etc.
- **Données globales**
  - se composant des variables non localisées

Les zones pour les données globales et la mémoire du programme CEI peuvent être synchronisées l'une par rapport à l'autre.

Synchronisation des zones pour les données globales et la mémoire du programme CEI (Momentum 171 CCS 760 00-CEI) :



Taille de la zone de la mémoire du programme CEI

La configuration pour la zone du programme utilisateur CEI se fait sous **En ligne** → **Statistiques mémoire...** → **Statistiques mémoire** dans la zone de texte **Configurées**. La configuration n'est possible que si le PC et l'automate sont hors ligne. Si vous n'utilisez pas ou peu de variables non localisées, et faites de même pour les liaisons modules, vous pouvez définir une taille importante de la mémoire programme CEI car vous ne devez pratiquement pas laisser de mémoire pour les données globales.

Taille de la zone pour les données globales

La zone pour les données globales (variables non localisées et liaisons de module) se calcule avec la formule suivante :

Zone pour les données globales = taille de la mémoire de l'instruction chargeable - mémoire du programme CEI

L'occupation actuelle des différentes zones (EFB, données d'instance, programme utilisateur, etc.) est affichée sous **En ligne** → **Statistiques mémoire...** → **Statistiques mémoire**. Cet affichage n'est possible que si le PC et l'automate sont en ligne.

Message d'erreur lors du chargement du programme

Si vous obtenez un message d'erreur lors du chargement du programme dans l'automate vous indiquant que le programme utilisateur est trop important pour la mémoire de l'automate, il peut y avoir deux raisons :

1. La mémoire est véritablement trop petite.
2. La zone pour les données globales et la zone de mémoire du programme CEI ne sont pas synchronisées de manière optimale l'une par rapport à l'autre (voir le présent chapitre).

6.7 Optimisation de mémoire pour les UC Atrium

Introduction

Aperçu

Ce chapitre décrit l'optimisation de mémoire pour les UC Atrium.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Généralités sur l'optimisation de la mémoire pour les UC Atrium	206
Utilisation de CEI	207
Synchronisation des zones pour les données globales et la mémoire du programme CEI (Atrium)	208

Généralités sur l'optimisation de la mémoire pour les UC Atrium

Mémoire logique

La zone de la mémoire du programme qui accueille votre programme utilisateur est appelée zone de logique. Cette zone détermine également la taille maximale de votre programme utilisateur.

La taille actuelle de la zone de logique peut être visualisée sous **Projet** → **Configurateur automate** dans la Vue d'ensemble de la configuration dans la zone **Automate**. La taille de la mémoire est affichée en ko pour CEI.

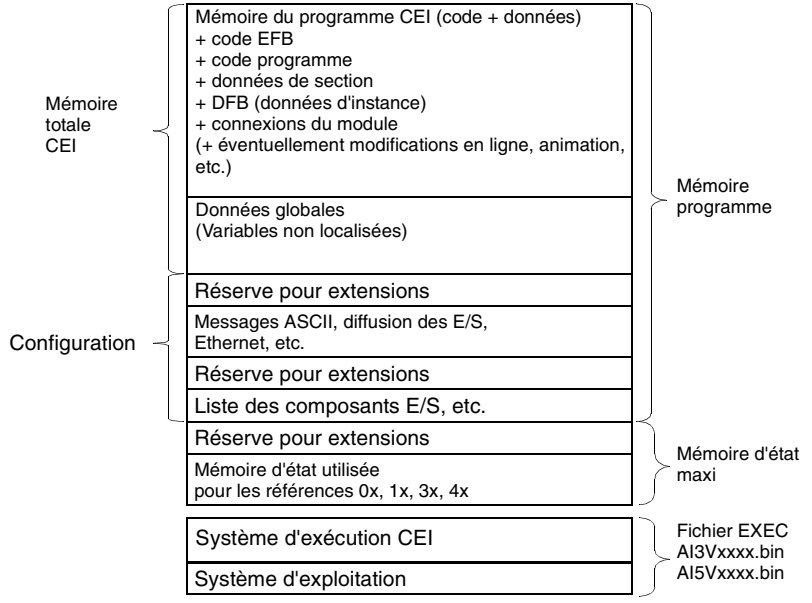
Optimisation de la mémoire logique

Vous avez différentes possibilités pour optimiser la mémoire logique en fonction de vos besoins :

- *Utilisation de CEI, p. 207*
- *Synchronisation des zones pour les données globales et la mémoire du programme CEI (Atrium), p. 208*

Note : Prenez également en compteles possibilités d'optimisation de la mémoire indépendantes de l'automate (voir *Remarques générales sur l'optimisation de la mémoire, p. 153*).

Structure de la mémoire d'une UC Atrium (représentation simplifiée) :



Utilisation de CEI

Introduction

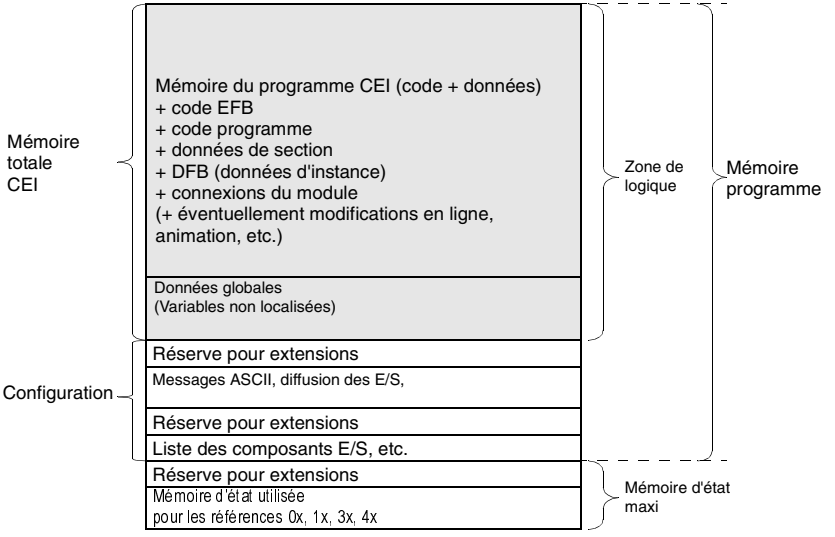
Les fichiers EXEC nécessaires à la famille d'UC Atrium contiennent les systèmes d'exploitation pour CEI (voir également *Introduction à l'installation*).

Lors de l'utilisation d'Atrium 180 CCO 121 01 vous chargez le fichier EXEC "AI3Vxxxx.bin".

Lors de l'utilisation d'Atrium 180 CCO 241 01et 180 CCO 241 11 vous chargez le fichier EXEC "AI5Vxxxx.bin".

Sélectionnez l'entrée **Validée** sous **Projet** → **Configuration automate** → **Sélection automate** dans la zone de liste **Moteur d'exécution** et déplacez le curseur **Taille du tas CEI** jusqu'à la bordure droite (valeur maximale). Ainsi, la zone LL984 est entièrement désactivée et vous disposez alors de toute la zone de logique pour votre programme utilisateur CEI.

Création de la mémoire UC Atrium lors d'une utilisation exclusive de CEI :



Message d'erreur lors du chargement du programme

Si vous obtenez un message d'erreur lors du chargement vous indiquant que le programme utilisateur est trop important pour la mémoire de l'automate, il peut y avoir trois raisons :

1. La mémoire est véritablement trop petite.
2. La zone de logique est trop petite (voir le présent chapitre).
3. La zone pour les données globales et la mémoire du programme CEI ne sont pas synchronisées de manière optimale l'une par rapport à l'autre (voir le chapitre *Synchronisation des zones pour les données globales et la mémoire du programme CEI (Atrium), p. 208*).

Synchronisation des zones pour les données globales et la mémoire du programme CEI (Atrium)

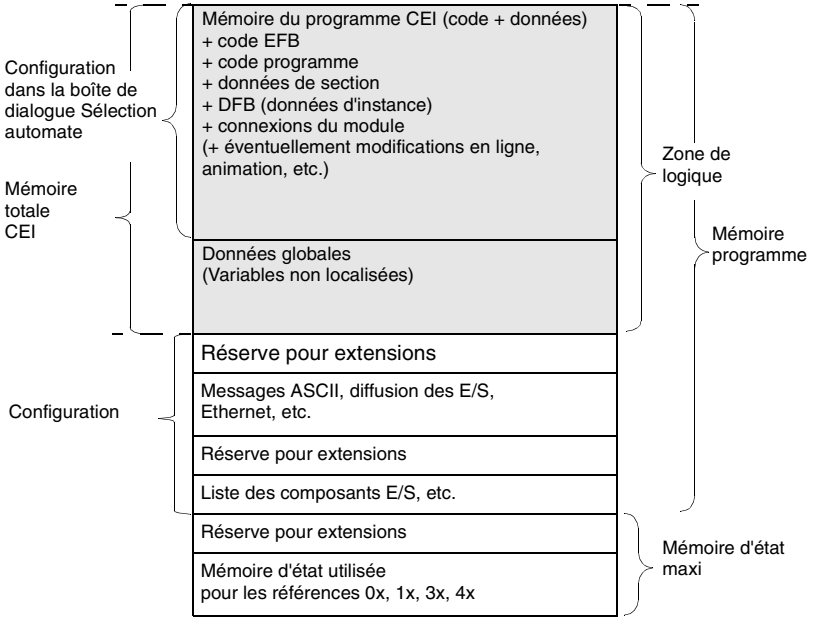
Introduction

La mémoire totale CEI définie (voir chapitre *Utilisation de CEI, p. 207*) se compose de 2 zones.

- **Mémoire du programme CEI**
  - se composant des codes EFB,
  - des codes programme,
  - des données de section,
  - des données d'instance DFB,
  - des liaisons de module,
  - éventuellement des données des modifications en ligne,
  - éventuellement des données d'animation, etc.
- **Données globales**
  - se composant des variables non localisées

Les zones pour les données globales et la mémoire du programme CEI peuvent être synchronisées l'une par rapport à l'autre.

Synchronisation des zones pour les données globales et la mémoire du programme CEI (Atrium) :


Configuration dans la boîte de dialogue Sélection automate
Mémoire totale CEI
Mémoire du programme CEI (code + données)
+ code EFB
+ code programme
+ données de section
+ DFB (données d'instance)
+ connexions du module
(+ éventuellement modifications en ligne, animation, etc.)
Données globales
(Variables non localisées)
Zone de logique
Mémoire programme
Configuration
Réserve pour extensions
Messages ASCII, diffusion des E/S, Ethernet, etc.
Réserve pour extensions
Liste des composants E/S, etc.
Réserve pour extensions
Mémoire d'état utilisée pour les références 0x, 1x, 3x, 4x
Mémoire d'état maxi

Taille de la zone de la mémoire du programme CEI

La configuration de la Mémoire du programme CEI se fait sous **Projet** → **Configuration automate** → **Sélection automate** dans la zone **CEI**. Indiquez ici la taille de la mémoire totale CEI et des données globales afin d'obtenir la taille de la mémoire du programme CEI (mémoire du programme CEI = mémoire totale CEI - données globales). La configuration n'est possible que si le PC et l'automate sont hors ligne. Si vous n'utilisez pas ou peu de variables non localisées, et faites de même pour les liaisons modules, vous pouvez définir une taille importante de la mémoire programme CEI car vous ne devez pratiquement pas laisser de mémoire pour les données globales.

Taille de la zone pour les données globales

La zone pour les données globales (variables non localisées) se calcule avec la formule suivante :

Zone pour les données globales = taille de la mémoire de l'instruction chargeable - mémoire du programme CEI

L'occupation actuelle des différentes zones (EFB, données d'instance, programme utilisateur, etc.) est affichée sous **En ligne** → **Statistiques mémoire...** → **Statistiques mémoire**. Cet affichage n'est possible que si le PC et l'automate sont en ligne.

Message d'erreur lors du chargement du programme

Si vous obtenez un message d'erreur lors du chargement du programme dans l'automate vous indiquant que le programme utilisateur est trop important pour la mémoire de l'automate, il peut y avoir trois raisons :

1. La mémoire est véritablement trop petite.
2. La taille de la mémoire totale CEI est trop petite (voir le chapitre *Utilisation de CEI, p. 207*).
3. La zone pour les données globales et la zone de mémoire du programme CEI ne sont pas synchronisées de manière optimale l'une par rapport à l'autre (voir le présent chapitre).

Langage des blocs fonctions FDB

7

Introduction

Aperçu

Ce chapitre décrit le langage des blocs fonctions FDB conforme à la norme CEI-1131.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
7.1	Remarques générales sur le langage du bloc fonction FDB	213
7.2	Objets du langage de bloc fonction FBD	214
7.3	Tâches exécutées en langage de bloc fonction FBD	223
7.4	Génération de codes dans le cadre du langage de bloc fonction FBD	229
7.5	Fonctions en ligne du langage de bloc fonction FBD	231
7.6	Elaboration d'un programme avec le langage de bloc fonction FBD	233

7.1 Remarques générales sur le langage du bloc fonction FDB

Remarques générales sur le langage du bloc fonction FDB

Introduction

Les objets du langage de programmation FBD (Diagramme de blocs fonctionnels) offrent des aides permettant de structurer une section en un ensemble de :

- EFB (fonctions élémentaires et blocs fonctions élémentaires) (voir *EFB, p. 215*),
- DFB (blocs fonctions dérivés) (voir *DFB, p. 217*) et
- UDEFB (fonctions et blocs fonctions définis par l'utilisateur) (voir *UDEFB, p. 218*).

Ces objets, regroupés sous l'abréviation générique FFB, peuvent être liés les uns aux autres par :

- des liaisons (voir *Liaison, p. 219*) ou
- de paramètres réels (voir *Paramètres réels, p. 220*).

Vous pouvez également insérer une logique étendue sous forme de macros dans votre section FBD (voir également *Macros, p. 535*).

Chaque section peut, théoriquement, contenir un nombre quelconque de FFB, et donc un nombre quelconque d'entrées et de sorties. Il est toutefois judicieux de diviser un programme global en différentes unités logiques, c'est-à-dire en différentes sections.

La logique de la section peut être commentée par des objets texte (voir *Objet texte, p. 222*).

Ordre d'exécution

L'ordre d'exécution des différents FFB dans une section FBD est déterminé par le flux de données à l'intérieur de la section (voir également *Ordre d'exécution des FFB, p. 226*).

Editer avec le clavier

Normalement, dans Concept, l'édition s'effectue à l'aide de la souris, mais est également possible à l'aide du clavier (voire également *Touches courtes dans l'éditeur FBDet SFC, p. 874*).

Conformité CEI

Description de la conformité CEI du langage de programmation FBD, voir *Conformité CEI, p. 887*.

7.2 Objets du langage de bloc fonction FBD

Introduction

Aperçu

Ce chapitre décrit les objets du langage de bloc fonction FBD.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Fonctions et blocs fonctions (FFB)	215
Liaison	219
Paramètres réels	220
Objet texte	222

Fonctions et blocs fonctions (FFB)

Introduction

FFB est le terme générique pour :
- EFB (fonctions élémentaires et blocs fonctions élémentaires), (voir *EFB, p. 215*)
- DFB (blocs fonctions dérivés) (voir *DFB, p. 217*)
- UDEFB (fonctions élémentaires dérivées et blocs fonctions élémentaires dérivés), (voir *UDEFB, p. 218*)

EFB

EFB est le terme générique pour :
- Fonction élémentaire (voir *Fonction élémentaire, p. 215*)
- Bloc fonction élémentaire (voir *Bloc fonction élémentaire, p. 216*)

Les EFB sont des fonctions et de blocs fonctions que Concept met à votre disposition sous forme de bibliothèques. La logique des EFB est établie en langage de programmation C et ne peut pas être modifiée dans l'éditeur FBD.

Fonction élémentaire

Les fonctions n'ont pas d'état interne. Pour des valeurs égales aux entrées, la valeur à la sortie est la même pour toutes les exécutions de la fonction. Par exemple, l'addition de deux valeurs livre le même résultat à chaque exécution de la fonction.

Une Fonction élémentaire est représentée graphiquement comme un cadre avec des entrées et une sortie. Les entrées sont toujours représentées à la gauche et les sorties toujours à la droite du cadre. Le nom de la fonction, c'est-à-dire le type de fonction, est affiché au centre du cadre. Le numéro de fonction est affiché au-dessus du cadre.

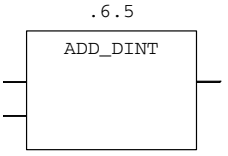
Le numéro de fonction ne peut pas être modifié et présente toujours la structure .n.m.

.n = numéro courant de la section

.n = numéro courant de la fonction

Les fonctions ne sont exécutées pour FDB que lorsque l'entrée EN=1 ou lorsque l'entrée EN est désactivée (voir également *EN et ENO, p. 218*).

Fonction élémentaire



Bloc fonction élémentaire

Les blocs fonctions ont des états internes. Pour des valeurs égales aux entrées, la valeur à la sortie est toujours différente à chaque exécution du bloc fonction. Par exemple, pour un compteur, la valeur à la sortie augmente.

Un bloc fonction est représenté graphiquement comme un cadre avec des entrées et des sorties. Les entrées sont toujours représentées à la gauche et les sorties toujours à la droite du cadre. Le nom du bloc fonction, c'est-à-dire le type bloc fonction, est affiché au centre du cadre. Le nom d'instance est affiché au-dessus du cadre. Le nom d'instance permet d'identifier précisément le bloc fonction dans un projet.

Le nom d'instance est généré automatiquement et présente la structure suivante : FBI_n_m

FBI = Instance de bloc fonction

n = Numéro de section (numéro courant)

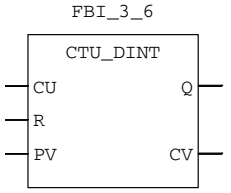
m = Numéro de l'objet FFB dans la section (numéro courant)

Le nom d'instance peut être édité par le biais de la boîte de dialogue **Objet** → **Caractéristiques** du bloc fonction. Le nom d'instance doit être précis dans tout le projet sans égard aux majuscules et minuscules. Si le nom entré existe déjà, vous en êtes averti et vous devez choisir un autre nom. Le nom d'instance doit correspondre aux conventions de nom CEI, sinon un message d'erreur s'affiche.

Note : Selon CEI1131-3 seules des lettres sont autorisées comme premier caractère des noms d'instances. Si vous souhaitez cependant utiliser également des chiffres comme premiers caractères, vous pouvez le faire avec la commande **Options** → **Environnement** → **Extensions CEI** → **Chiffres en tête de qualificatif autorisés** Libérer.

Les blocs fonction ne sont exécutés pour FDB que lorsque l'entrée EN=1 ou lorsque l'entrée EN est désactivée (voir également *EN et ENO, p. 218*).

Bloc fonction élémentaire



DFB

Les blocs fonctions (DFB) dérivés sont des blocs fonctions que vous avez définis dans Concept-DFB.

Dans les DFB, on ne distingue pas la fonction du bloc fonction. Indépendamment de leur structure interne, elles sont toujours traitées comme des blocs fonctions.

Un DFB est représenté graphiquement comme un cadre avec deux lignes verticales et des entrées et des sorties. Les entrées sont toujours représentées à la gauche et les sorties toujours à la droite du cadre. Le nom du DFB est affiché au centre du cadre. Le nom d'instance est affiché au-dessus du cadre. Le nom d'instance permet d'identifier précisément le bloc fonction dans un projet.

Le nom d'instance est généré automatiquement et présente la structure suivante : FBI_n_m

FBI = Instance de bloc fonction

n = Numéro de section (numéro courant)

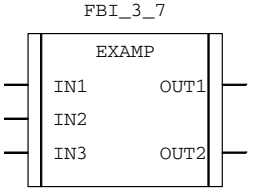
m = Numéro de l'objet FFB dans la section (numéro courant)

Le nom d'instance peut être édité par le biais de la boîte de dialogue **Objet** → **Caractéristiques** du DFB. Le nom d'instance doit être précis dans tout le projet sans égard aux majuscules et minuscules. Si le nom entré existe déjà, vous en êtes averti et vous devez choisir un autre nom. Le nom d'instance doit correspondre auxconventions de nom CEI, sinon un message d'erreur s'affiche.

Note : Selon CEI1131-3 seules des lettres sont autorisées comme premier caractère des noms d'instances. Si vous souhaitez cependant utiliser également des chiffres comme premiers caractères, vous pouvez le faire avec la commande **Options** → **Environnement** → **Extensions CEI** → **Chiffres en tête de qualificatif autorisés** Libérer.

Les blocs fonction dérivés ne sont exécutés pour FDB que lorsque l'entrée EN=1 ou lorsque l'entrée EN est désactivée (voir également *EN et ENO, p. 218*).

Bloc fonction dérivé



UDEFB

UDEFB est le terme générique pour :

- Fonction élémentaire définie par l'utilisateur
- Bloc fonction élémentaire défini par l'utilisateur

Les UDEFB sont des fonctions et blocs fonctions que vous avez programmés avec Concept-EFB en langage de programmation C++ et que vous mettez sous forme de bibliothèques à la disposition de Concept.

Il n'y a pas de différence fonctionnelle entre les UDEFB et les EFB dans Concept.

EN et ENO

Pour tous les FFB, une entrée EN et une sortie ENO peuvent être configurées.

La configuration d'EN et d'ENO est activée ou désactivée dans la boîte de dialogue des **Propriétés FFB** . La boîte de dialogue est appelée par la commande **Objets** → **Caractéristiques...** ou en cliquant deux fois sur le FFB.

Au cas où la valeur d'EN est égale à "0", lorsque le FFB est appelé, les algorithmes définis par le FFB ne seront pas exécutés et les sorties conserveront leur valeur précédente. Dans ce cas, la valeur d'ENO sera automatiquement placée à "0".

Au cas où la valeur d'EN est égale à "1", lorsque le FFB est appelé, les algorithmes définis par le FFB sont exécutés. Après l'exécution exempte d'erreur de ces algorithmes, la valeur d'ENO est automatiquement placée à "1". Si une erreur se produit durant l'exécution de ces algorithmes, ENO est placée à "0".

Le comportement de sortie des FFB pour FBD est indépendant du fait que les FFB soient appelés sans EN/ENO ou avec EN=1.

Liaison

Description	Les connexions sont des liens entre des FFB. Plusieurs liaisons peuvent être reliées à une sortie FFB. Les points de liaison sont marqués par un cercle rempli.
Type de données	Les types de données respectifs des entrées/sorties à relier doivent correspondre les uns aux autres
Créer des connexions	Des liaisons peuvent être générées par **Objets** → **Liaison** .
Traitement des liaisons	Les liaisons peuvent être traitées en mode sélection. Le chevauchement avec d'autres objets est admis.
Configuration de boucles	Les boucles ne peuvent pas être configurées par le biais de liaisons, étant donné que, dans ce cas, l'ordre d'exécution dans la section ne peut pas être défini de façon unique. Les boucles doivent être résolues par le biais de paramètres réels (voir *Configuration de boucles, p. 228*).

Paramètres réels

Introduction

Durant l'exécution du programme, les valeurs sont transmises, par le biais de paramètres réels, du processus ou d'autres paramètres réels au FFB, et renvoyées à nouveau à la sortie après le traitement.

Ces paramètres réels peuvent être les suivants :
- Adresse directe (voir *Adresses directes, p. 53*)
- Variables localisées (voir *Variables :, p. 50*)
- Variable non-localisée (voir *Variables :, p. 50*)
- Constantes (voir *Constantes, p. 51*)
- Littéraux (voir *Littéraux (valeurs), p. 52*)

Adresses directes

L'indication/l'affichage des adresses directes peut s'effectuer dans différents formats. La définition du format d'affichage s'effectue dans le dialogue **Options** → **Environnement** → **Commun...**. La définition du format d'affichage n'a aucun impact sur le format d'entrée, ceci signifie que l'entrée d'adresses directes peut s'effectuer dans un format quelconque.

Les formats d'adresses suivants sont disponibles :
- **Format standard (400001)**
  L'adresse à cinq positions intervient directement après le premier chiffre (de la Référence).
- **Format à séparateur (4 : 00001)**
  Le premier chiffre (la Référence) est séparé(e) par un point double (:) de l'adresse à cinq positions.
- **Format compact (4 : 1)**
  Le premier chiffre (la Référence) est séparé(e) par un point double (:) de l'adresse suivante, les zéros en tête n'étant pas indiqués dans l'adresse .
- **Format CEI (QW1)**
  Un identificateur est placé en première position, après CEI, suivi de l'adresse à cinq positions.
  - %0x12345 = %Q12345
  - %1x12345 = %I12345
  - %3x12345 = %IW12345
  - %4x12345 = %QW12345

Type de données

Le type des données du paramètre réel doit correspondre au type des données de l'entrée/la sortie. La seule exception concerne les entrées/sortie génériques dont le type de données est déterminé par le paramètre formel. On choisira un type de données adapté pour le bloc fonction, si tous les paramètres réels sont constitués de libellés.

Valeurs initiales

Les FFB utilisant aux entrées des paramètres réels, auxquels aucune valeur n'a encore été affectée, fonctionnent avec les valeurs initiales de ces paramètres réels.

Entrées non reliées

Note : Des entrées FFB non-liées sont équipées standard de la valeur " 0 ".

Objet texte

Introduction

Dans le langage blocs fonctions FBD, les textes peuvent être placés sous forme d'objets texte. La taille de ces objets texte est fonction de la longueur du texte. Selon la longueur du texte, la taille de l'objet peut être agrandie, dans les sens vertical et horizontal, d'unités de grille supplémentaires. Les objets texte ne doivent pas se chevaucher avec des FFB, le chevauchement avec des liaisons est toutefois admis.

Capacité de mémoire

Les objets texte n'occupent pas d'espace de mémoire sur l'API, étant donné que le texte n'est pas chargé sur l'API.

7.3 Tâches exécutées en langage de bloc fonction FBD

Introduction

Aperçu

Ce chapitre décrit ce que l'on peut faire avec le langage de bloc fonction FBD.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Placement de fonctions et de blocs fonctions	224
Ordre d'exécution des FFB	226
Configuration de boucles	228

Placement de fonctions et de blocs fonctions

Sélection de FFB

Avec **Objets** → **Sélection FFB...** vous pouvez ouvrir un dialogue pour la sélection de FFB. Cette boîte de dialogue est modeless, c.-à-d. qu'elle ne se referme pas automatiquement après le lacement d'un FFB, mais reste ouverte jusqu'à ce que vous la fermiez. Si vous avez ouvert plusieurs sections FBD et appelé la boîte de dialogue, une seule boîte de dialogue est ouverte et disponible pour toutes les sections. Pour toutes les autres sections (pas l'éditeur FBD),la boîte de dialogue n'est pas disponible. Lorsque les sections FBD sont transformées en symboles (minimaliser la fenêtre), la boîte de dialogue se ferme. Lorsque l'un des symboles de section FBD est à nouveau appelé, la boîte de dialogue s'ouvre à nouveau automatiquement .

Au premier démarrage de Concept, l'affichage des FFB s'oriente sur les bibliothèques. Ceci signifie que pour sélectionner un FFB, il faut d'abord sélectionner la bibliothèque correspondante à l'aide du bouton de commande **Bibliothèque**. Vous pouvez ensuite sélectionner le groupe correspondant dans la zone de liste **Groupe**. Vous pouvez à présent sélectionner le FFB souhaité dans la zone de liste Type FFB.

Au cas où vous ne savez pas dans quelle bibliothèque/ dans quel groupe le FFB souhaité se trouve, vous pouvez utiliser le bouton de commande **Orienté FFB...**, pour appeler un dialogue orienté sur les FFB. Ce dialogue propose une liste triée par ordre alphabétique de tous les FFB de toutes les bibliothèques et tous les groupes.

Après chaque démarrage ultérieur du projet, la vue que vous avez sélectionnée apparaît.

Après la sélection du FFB, choisissez sa position dans la section. Le symbole curseur se place comme un petit FFB à côté du symbole des FFB et la croix indique sa position (le coin gauche supérieur des FFB). Par un clic avec la touche gauche de la souris, vous placez le FFB.

Placement de FFB (fonctions et blocs fonctions)

Dans l'éditeur de langage de bloc fonction FBD, la fenêtre est placée sur une grille logique. Les FFB (voir *Fonctions et blocs fonctions (FFB), p. 215*) sont ajustés sur le quadrillage de cette grille lors du placement. Lorsque les FFB sont placés hors du cadre de la section ou en cas de chevauchement avec un autre FFB, un message d'erreur est déclenché, et le FFB n'est pas placé. Lors du placement à une entrée/sortie FFB, les paramètres réels peuvent se chevaucher avec un autre objet, mais ils ne doivent pas franchir les limites du cadre de section. Lorsqu'une liaison est utilisée pour établir une liaison à un autre FFB, cette liaison est contrôlée. Si cette liaison n'est pas permise, vous obtenez un message et la liaison n'est pas générée. Lors de la génération de liaisons, des chevauchements et croisements avec d'autres liaisons et FFB sont permis. Lorsqu'un FFB est sélectionne, ce commentaire est affiché dans la première colonne de la ligne d'état. Lorsqu'un paramètre réel est sélectionné, son nom et – le cas échéant – son adresse, son affectation des E/S et son commentaire apparaissent dans la première colonne de la ligne d'état.

Modifier type FFB

Avec la commande **Objets** → **Remplacer FFB...** vous pouvez remplacer des FFB déjà placés dans la section par des FFB d'un autre type (par ex. un AND par un OR). Cependant, les variables attribuées aux FFB restent préservées lorsque le type de données et la position des entrées/sorties de 'l'ancien'et du nouveau FFB correspondent.

Note : Des FFB aux entrées/sorties de données du type ANY (des FFB génériques) ne peuvent être remplacés.

Ordre d’exécution des FFB

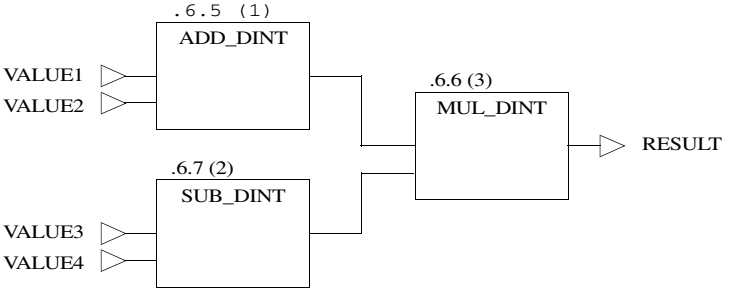
Introduction

L’ordre d’exécution est d’abord déterminé par l’ordre observé lors du placement des FFB. Lorsque, par la suite, les FFB sont reliés par des liaisons graphiques, l’ordre d’exécution est alors déterminé par le flux de données.

Afficher ordre d’exécution FFB

Avec la commande **Objets** → **Afficher ordre d’exécution FFB**, vous pouvez afficher l’ordre d’exécution des FFB. L'ordre d'exécution des FFB est indiqué par l'affichage du numéro d'exécution (numéro entre parenthèses après le nom d’instance ou le numéro de fonction).

Affichage de l'ordre d'exécution des FFB



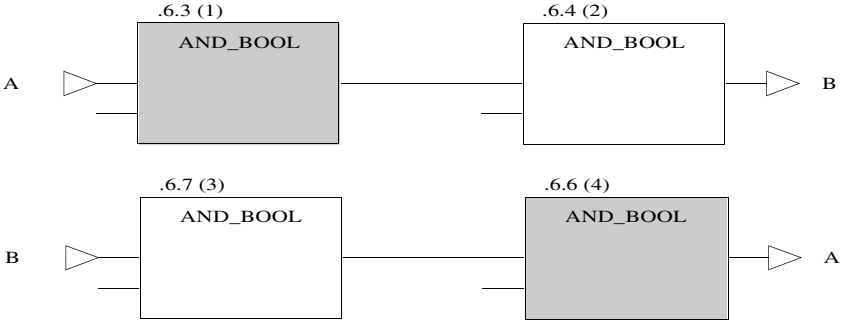
Modifier ordre d’exécution FFB

L’ordre d’exécution peut être modifié ultérieurement de façon ciblée à l’aide de la commande **Objets** → **Modifier ordre d’exécution FFB**, uniquement cependant lorsque la règle du flux de donnée n’est pas violée.

Modification de l'ordre d'exécution de deux réseaux (dans une boucle)

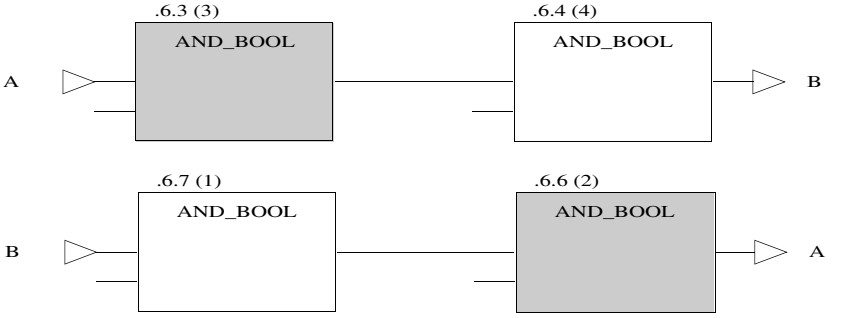
Cette modification ne peut être effectuée que si les deux FFB sont liés par la variable d'asservissement de la boucle.

Etape 1 : sélectionnez les deux FFB.



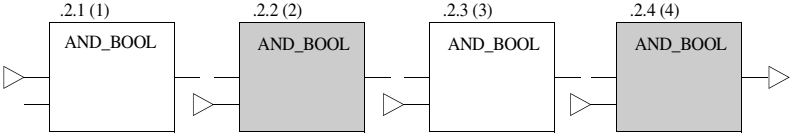
Etape 2 : sélectionnez la commande **Modifier ordre d'exécution FFB**.

Réaction : l'ordre d'exécution a été modifié comme suit :

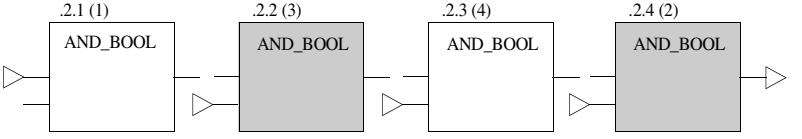


Modification de l'ordre d'exécution de FFB qui sont exécutés selon l'ordre de placement

L'opération de modification permet la création d'un autre ordre souhaité (dans certains cas via l'attribution étape par étape de plusieurs FFB).



Réaction : l'ordre d'exécution a été modifié comme suit :

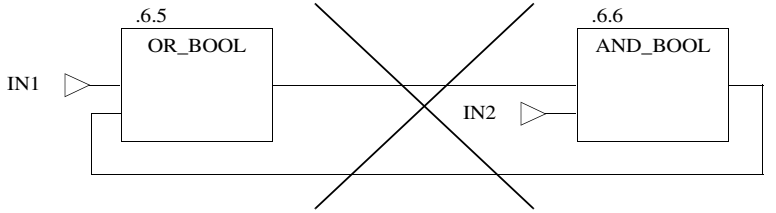


Configuration de boucles

Boucles non permises

La configuration de boucles exclusivement par le biais de liaisons n'est pas permise, étant donné que, dans ce cas, une détermination unique du flux de données n'est pas possible (la sortie d'un FFB est l'entrée du FFB suivant, et la sortie de celui-ci est à son tour l'entrée du premier).

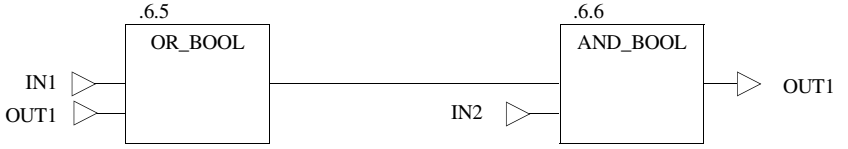
Boucles non permises par le biais de liaisons



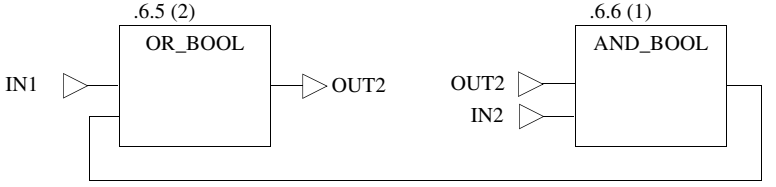
Résolution par le biais d'un paramètre réel

Une telle logique doit être résolue par le biais de paramètres réels, afin que le flux de données puisse être défini de façon unique .

Boucle résolue par le biais d'un paramètre réel : Variante 1



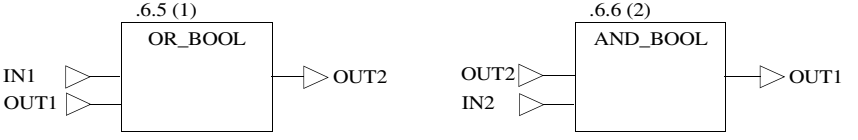
Boucle résolue par le biais d'un paramètre réel : Variante 2



Résolution par le biais de plusieurs paramètres réels

Les boucles résolues par le biais de plusieurs paramètres réels sont également permises. Pour de telles boucles, l'ordre d'exécution peut être influencé ultérieurement par une exécution – éventuellement répétée – de la commande **Objets** → **Modifier ordre d'exécution des FFB** (voir également *Ordre d'exécution des FFB, p. 226*).

Boucle par le biais de plusieurs paramètres réels



7.4 Génération de codes dans le cadre du langage de bloc fonction FBD

Options de génération de code

Introduction

Avec la commande **Projet** → **Options de génération de code**,vous pouvez définir les Options de génération de code.

Des données de diagnostic comprennent

Si vous activez la case à cocher **Inclure les données de diagnostic**, des informations pour le diagnostic de processus seront générées en plus lors de la génération de codes (par ex. Diagnostic de transition (voir *Diagnostic de transition, p. 323*), Codes de diagnostic pour des blocs fonctions avec diagnostic étendu, comme par exemple XACT, XLOCK, etc.). L'évaluation de ce diagnostic de processus peut s'effectuer avec MonitorPro ou FactoryLink, par exemple.

Code le plus rapide (vérification limitée)

Si vous activez la case à cocher **Code le plus rapide (vérification limitée)**, un code à temps optimisé sera généré. L'optimisation du temps d'exécution est obtenue de la façon suivante : l'arithmétique sur les entiers (par ex. "+" ou "-") est réalisée via des commandes simples de processeur, au lieu d'être réalisée par des appels de EFB.

Les commandes de processeur sont bien plus rapides que les appels de EFB, mais ils ne génèrent pas de messages d'erreur, comme Dépassement en arithmétique ou Débordement de tableau (d'Array). Vous ne devez utiliser cette option que si vous vous êtes assuré que le programme ne contient aucune erreur d'arithmétique.

Si **Code le plus rapide (vérification limitée)** a été sélectionné, l'addition IN + 1 est exécutée par la commande de processeur "add". Le code est plus rapide que lorsque l'EFB ADD_INT est appelé. Mais aucune erreur d'exécution n'est générée si la valeur "IN1" est égale à 32767 . Dans ce cas, "OUT1" déborderait de 32767 à -32768.

7.5 Fonctions en ligne du langage de bloc fonction FBD

Fonctions en ligne

Introduction

Deux modes d'animation sont disponibles dans l'éditeur FBD :
- Animation de variables et liaisons binaires
- Animation d'objets sélectionnés

Ces modes sont également disponibles lors de l'affichage d'une instance de DFB (Bouton de commande **Détail...** dans la boîte de dialogue **Bloc fonction : xxx**).

Note : Lorsque la section animée est utilisée comme section de transition pour SFC et que la transition (et donc la section de transition) n'est pas traitée, l'état **INVALIDE** apparaît dans la section de transition animée.

Animation de variables et liaisons binaires

L'animation de variables et liaisons binaires est activée à l'aide de la commande **En ligne** → **Animer valeurs binaires** .

Dans ce mode, l'état de signal actuel des variable binaires, des adresses directes des domaines 0x et 1x et des liaisons binaires est affiché dans la fenêtre de l'éditeur.

Animation d'objets sélectionnés

L'animation d'objets sélectionnés est activée à l'aide de la commande **En ligne** → **Animer sélection**.

Dans ce mode, l'état de signal actuel des liaisons, variables, variables multi-éléments et littéraux sélectionnés est affiché dans la fenêtre de l'éditeur.

Note : Si vous voulez animer toutes les variables/liaisons de la section, vous pouvez utiliser **CTRL**+**A**, pour marquer la section entière, puis **En ligne** → **Animer sélection** (**CTRL**+**W**) pour animer toutes les variables et liaisons de la section.

Si vous sélectionnez une valeur numérique à une entrée/sortie, le nom de la variable, son adresse directe et son affectation E/S (si elle est disponible) ainsi que son commentaire s'affichent dans la barre d'état.

Note : Les objets sélectionnés continuent de l'être après une nouvelle sélection d'Animer sélection, afin de les conserver pour une lecture ultérieure ou pour permettre une modification aisée de la liste des objets.

Signification des couleurs

12 jeux de couleurs différents sont disponibles à des fins d'animation. Vous trouverez une vue d'ensemble des jeux de couleurs et de leur signification dans l'aide en ligne (astuce : recherchez dans l'aide en ligne en entrant "Couleurs" dans l'index).

7.6 Elaboration d'un programme avec le langage de bloc fonction FBD

Etablissement d'un programme dans le langage de bloc fonction FBD

Introduction

La description suivante donne un exemple pour l'établissement d'un programme dans le langage de bloc fonction (FBD). L'établissement d'un programme dans le langage de bloc fonction FBD s'effectue en deux étapes principales :

Etape	Action
1	Génération d'une section (voir *Génération d'une section, p. 233*)
2	Elaboration de la logique (voir *Elaboration de la logique, p. 234*)

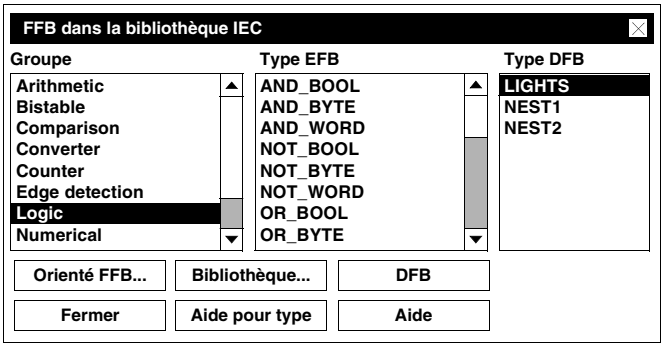
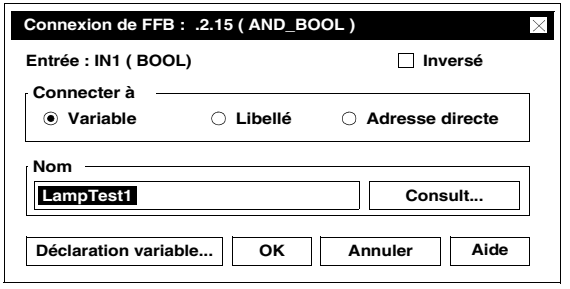
Génération d'une section

Pour générer une section, effectuez les opérations suivantes :

Etape	Action
1	Générez avec la commande **Fichier** → **Nouvelle section...** une nouvelle section et attribuez-lui un nom de la section. **Note :** Le nom de la section (au max. 32 caractères) doit être unique dans tout le projet, aucune différence n'étant faite entre l'écriture en lettres majuscules et en minuscules. Si le nom de la section entré existe déjà, vous recevez un avertissement et devez sélectionner un autre nom. Le nom de la section doit correspondre aux conventions de nom CEI, sinon un message d'erreur s'affiche. **Note :** Selon CEI 1131-3, uniquement des lettres sont permises comme premier caractère de noms. Si vous souhaitez cependant utiliser également des chiffres comme premiers caractères, vous pouvez le faire avec la commande **Options** → **Environnement** → **Extensions CEI** → **Chiffres en tête de qualificatif autorisés** Libérer.

Elaboration de la logique

Pour élaborer une logique, effectuez les opérations suivantes :

Etape	Action
1	Pour insérer un FFB dans la section, sélectionnez la commande **Objets** → **Sélection FFB...**. **Réaction :** La boîte de dialogue FFB de bibliothèque s'ouvre. 
2	Dans cette boîte de dialogue, vous pouvez employer le bouton de commande **Bibliothèque...** pour sélectionner une bibliothèque et y choisir un FFB. Mais vous pouvez également utiliser le bouton de commande **DFB** pour afficher les DFB que vous avez générés et en sélectionner un.
3	Placez uniquement le DFB sélectionné dans la section.
4	Lorsque vous avez placé tous les DFB sélectionnés, fermez la boîte de dialogue avec **Fermer**.
5	Activez le mode sélection avec **Objets** → **Mode sélection**, cliquez sur le FFB et décalez les FFB sur la position souhaitée.
6	Activez le mode de liaison avec **Objets** → **Liaison**, et établissez les liaisons entre les FFB.
7	A présent, activez à nouveau le mode sélection avec **Objets** → **Mode sélection**, et cliquez deux fois sur l'une des entrées/sorties non reliées. **Réaction :** La boîte de dialogue **Relier FFB** s'ouvre, dans laquelle vous pouvez affecter un paramètre réel à l'entrée/la sortie. 

Etape	Action

8	Selon la logique du programme, vous pouvez affecter à l'entrée/la sortie :
	• **Variable**
	  • Variable affectée
	  Par une variable localisée , vous pouvez affecter un signal d'entrée/de sortie de matériel du FFB à l'entrée/la sortie.
	  Le nom de la variable est affiché sur la fenêtre de l'éditeur à l'entrée/la sortie.
	  • Variable non-affectée
	  Vous pouvez utiliser la variable non localisée affectée à l'entrée/la sortie du FFB comme mémento, c'est-à-dire pour la résolution de boucles ou pour la transmission de valeur entre différentes sections.
	  Le nom de la variable est affiché sur la fenêtre de l'éditeur à l'entrée/la sortie.
	  • Constante
	  Vous pouvez affecter une constante à l'entrée du FFB. Vous pouvez transmettre la constante à d'autres sections. Vous déterminez la valeur des constantes dans l'éditeur des variables.
	  Le nom de la constante est affiché sur la fenêtre de l'éditeur à l'entrée.
	• **Libellé**
	  Vous pouvez affecter un libellé à l'entrée, c'est-à-dire que vous pouvez affecter directement une valeur à l'entrée/la sortie.
	  La valeur est affichée sur la fenêtre de l'éditeur à l'entrée.
	• **Adresse directe**
	  Par une adresse, vous pouvez affecter un signal d'entrée/de sortie de matériel à l'entrée/la sortie.
	  L'adresse est affichée sur la fenêtre de l'éditeur à l'entrée/la sortie.
	Note : Exemple d'appel de variables multi-éléments, voir *Appel de types de données dérivés, p. 603*.
	Note : De façon standard, les entrées FFB non reliées sont occupées par "0".

9	Sauvegardez à présent la section FBD avec la commande **Fichier** → **Enregistrer projet**.

Plan de contact LD

8

Introduction

Aperçu

Ce chapitre décrit le plan de contact LD conforme à la norme CEI-1131.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
8.1	Remarques générales sur le langage à contacts LD	239
8.2	Objets dans le langage à contacts LD	241
8.3	Tâches exécutées en langage à contacts LD	257
8.4	Génération de codes dans le cadre du langage à contacts LD	263
8.5	Fonctions en ligne dans le cadre du langage à contacts LD	265
8.6	Elaboration d'un programme avec le langage à contacts LD	268

8.1 Remarques générales sur le langage à contacts LD

Remarques générales sur le langage du schéma à contacts LD

Introduction

Le présent sous-chapitre décrit le schéma à contacts LD (Ladder Diagramm, schéma à contacts) selon CEI 1131-3.

La structure d'une section LD correspond à une voie de courant pour des montages à relais. Dans l'éditeur LD, la fenêtre est placée sur une grille logique présentant sur son côté gauche la barre d'alimentation gauche. Cette barre d'alimentation gauche correspond à la phase (conducteur L) d'une voie de courant. De même que sur une voie de courant, ne sont "traités", lors de la programmation, que les objets LD (contacts, bobines) qui sont branchés à une alimentation en courant, c'est-à-dire qui sont reliés à la barre d'alimentation gauche. La barre d'alimentation droite, qui correspond au conducteur neutre, n'est pas représentée. Cependant, toutes les bobines et sorties FFB y sont reliées en interne, ce qui permet d'établir un flux de courant.

Objets

Les objets du langage de programmation LD (schéma à contacts) proposent des aides permettant de diviser une section en un ensemble de :

- contacts (voir *Contacts, p. 242*),
- bobines (voir *Bobines, p. 244*) et
- FFB (fonctions et blocs fonctions) (voir *Fonctions et blocs fonctions (FFB), p. 247*).

Ces objets peuvent être reliés les uns aux autres par :

- des liaisons (voir *Liaison, p. 253*) ou
- des paramètres réels (voir *Paramètres réels, p. 254*).

Vous pouvez également insérer une logique étendue sous forme de macros dans votre section LD (voir également *Macros, p. 535*).

Chaque section peut, théoriquement, contenir un nombre quelconque d'objets, et donc un nombre quelconque d'entrées et de sorties. Il est toutefois conseillé de diviser un programme global en unités logiques, c'est-à-dire en différentes sections.

La logique de la section peut être commentée à l'aide d'objets texte (voir *Objet texte, p. 256*).

Ordre d'exécution

En principe, les sections LD sont traitées de haut en bas et de gauche à droite.

Les réseaux raccordés à la barre d'alimentation gauche sont traités de haut en bas.

L'ordre d'exécution des différents objets (contacts, bobines, FFB) au sein d'un réseau est déterminé par le flux de données.

Vous trouverez une description complète dans la section *Ordre d'exécution, p. 260*.

Editer à l'aide du clavier

Dans Concept, l'édition se fait normalement avec la souris, mais il est aussi possible d'utiliser le clavier (voir également *Touches courtes dans éditeur LD, p. 878*).

Pour simplifier l'édition avec le clavier, vous pouvez définir dans le fichier CONCEPT.INI (voir *Paramètres INI pour la section LD, p. 1157*) le nombre de colonnes autorisé par section avant qu'un saut de ligne automatique ne soit ajouté en cas d'extension d'une voie de courant. Lorsque la dernière colonne est atteinte, les objets qui suivent sont alors placés automatiquement dans la ligne suivante, à partir de la deuxième colonne. Les objets des différentes lignes sont reliés automatiquement, c'est-à-dire que les objets sont créés au sein d'une même voie de courant.

Conformité CEI

Pour obtenir la description de la conformité CEI du langage de programmation LD, voir *Conformité CEI, p. 887*.

8.2 Objets dans le langage à contacts LD

Introduction

Aperçu

Ce chapitre décrit les objets dans le langage à contacts LD.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Contacts	242
Bobines	244
Fonctions et blocs fonctions (FFB)	247
Liaison	253
Paramètres réels	254
Objet texte	256

Contacts

Introduction

Un contact est un élément LD qui transmet un état à la liaison horizontale à sa droite. Cet état résulte de la combinaison booléenne ET de l'état de la liaison horizontale sur sa partie gauche avec l'état de la variable/l'adresse directe associée.

Un contact ne change pas la valeur de la variable/l'adresse directe associée.

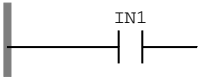
Les contacts suivants sont disponibles :

- Contact de fermeture (voir *Contact de fermeture, p. 242*)
- Contact de rupture (voir *Contact de rupture, p. 242*)
- Contact de reconnaissance de passages positifs (voir *Contacts de reconnaissance de passages positifs, p. 243*)
- Contact de reconnaissance de passages négatifs (voir *Contacts de reconnaissance de passages négatifs, p. 243*)

Contact de fermeture

Dans le cas de dispositifs de fermeture, l'état de la connexion de gauche est copié sur la connexion de droite, si l'état de la variable booléenne associée est ACT. Sinon, l'état de la connexion de droite est DESACT.

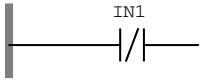
Contact de fermeture



Contact de rupture

Dans le cas de dispositifs d'ouverture, l'état de la connexion de gauche est copié sur la connexion de droite, si l'état de la variable booléenne associée est DESACT. Sinon, l'état de la connexion de droite est DESACT.

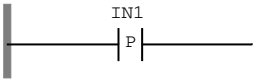
Contact de rupture



Contacts de reconnaissance de passages positifs

Dans le cas de contact destinés à la reconnaissance de passages positifs, la connexion de droite est ACT. pour un cycle de programme, si un passage de DESACT. à ACT. de la variable booléenne associée a lieu et qu'en même temps, l'état de la connexion de gauche est ACT. Sinon, l'état de la connexion de droite est DESACT.

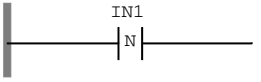
Contact de reconnaissance de passages positifs



Contacts de reconnaissance de passages négatifs

Dans le cas de contact destinés à la reconnaissance de passages négatifs, la connexion de droite est ACT. pour un cycle de programme, si un passage de ACT. à DESACT. de la variable booléenne associée a lieu et qu'en même temps, l'état de la connexion de gauche est ACT. Sinon, l'état de la connexion de droite est DESACT.

Contact de reconnaissance de passages négatifs



Bobines

Introduction

Une bobine est un élément LD qui transmet sans modification l'état de la liaison horizontale sur sa gauche à la liaison horizontale sur sa droite. Ceci étant, l'état est mémorisé dans la variable/l'adresse directe associée.

Comportement de démarrage des bobines

Lors de la procédure de reprise de l'API, on fait la différence entre la reprise à chaud et la reprise à froid :

- **Reprise à froid**
  Après une reprise à froid (Charger le programme avec **En ligne** → **Chargement**) tous les variables (de tous les types) seront mis sur "0" ou sur leurs valeurs initiales.
- **Reprise à chaud**
  Lors d'une reprise à chaud (arrêter et démarrer le programme) ou **En ligne** → **Chargement des modifications**) de différents procédures de départ existent pour les variables non-localisées/adresses directes et les variables non-localisées :
  - **Variables localisées /adresses directes**
    Lors d'une reprise à chaud, toutes les bobines (registres 0x) sont placées à "0" ou, si existant, à leur valeur initiale.
  - **Variables non-localisées**
    Lors d'une reprise à chaud, toutes les variables non-localisées conservent leur valeur actuelle (comportement mémorisant).

Ce comportement différent lors d'une reprise à chaud entraîne des particularités dans le comportement de reprise à chaud des objets LD "Bobine - Initialiser" et "Bobine - Réinitialiser". Le comportement de reprise à chaud est fonction du type de variables utilisé (comportement mémorisant, en cas d'utilisation de variables non-localisées; comportement non mémorisant en cas d'utilisation de variables/ d'adresses directes)

Au cas où vous avez besoin d'une bobine tamponnée, avec une variable localisée ou des adresses directes, utilisez les blocs fonctions RS ou SR de la bibliothèque de blocs CEI s.v.p.

Bobines disponibles

Les bobines suivantes sont disponibles :

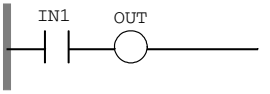
- Bobine (voir *Bobine, p. 245*)
- Bobine - Inversée (voir *Bobine – Inversée, p. 245*)
- Bobine - Initialiser (voir *Bobine – Initialiser, p. 246*)
- Bobine - Réinitialiser (voir *Bobine – Réinitialiser, p. 246*)
- Bobine – Front positif (voir *Bobine – Front positif, p. 245*)
- Bobine – Front négatif (voir *Bobine – Front négatif, p. 245*)

Bobine

Dans le cas de bobines, l'état de la connexion de gauche est copié sur la variable booléenne associée et sur la connexion de droite.

Normalement, les bobines viennent après des contacts ou des EFB, mais elles peuvent également être suivies par des contacts.

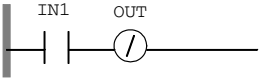
Bobine



Bobine – Inversée

Pour les bobines inversées, l'état de la liaison gauche est copié sur la liaison de droite. L'état inversé de la connexion de gauche est copié sur la variable booléenne associée. Si la connexion de gauche est DESACT., alors la connexion de droite est également DESACT. et la variable associée est ACT.

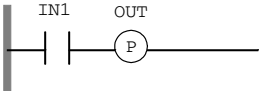
Bobine - Inversée



Bobine – Front positif

Dans le cas de bobines destinées à la reconnaissance de passages positifs, l'état de la connexion de gauche est copié sur la connexion de droite. La variable booléenne associée représente un cycle de programme ACT., si la connexion de gauche passe de DESACT. à ACT.

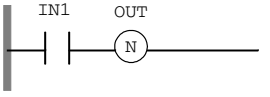
Bobine – Front positif



Bobine – Front négatif

Dans le cas de bobines destinées à la reconnaissance de passages négatifs, l'état de la connexion de gauche est copié sur la connexion de droite. La variable booléenne associée représente un cycle de programme ACT., si la connexion de gauche passe de ACT. à DESACT.

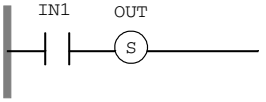
Bobine – Front négatif



Bobine – Initialiser

Dans le cas de "Définir bobines", l'état de la connexion de gauche est copié sur la connexion de droite. La variable booléenne associée est placée dans l'état ACT., si la connexion de gauche est dans l'état ACT. Sinon, elle reste inchangée. La variable booléenne associée peut être réinitialisée par "Réinitialiser bobines".

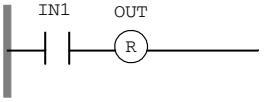
Bobine - Initialiser



Bobine – Réinitialiser

Dans le cas de " Réinitialiser bobines", l'état de la liaison de gauche est copié sur la liaison de droite. La variable booléenne associée est placée dans l'état DESACT., si la liaison de gauche est dans l'état ACT. Sinon, elle reste inchangée. La variable booléenne associée peut être initialisée par "Initialiser bobines".

Bobine - Réinitialiser



Fonctions et blocs fonctions (FFB)

Introduction

FFB est le terme générique pour :
- EFB (fonctions élémentaires et blocs fonctions élémentaires), (voir *EFB, p. 247*)
- DFB (blocs fonctions dérivés) (voir *DFB, p. 250*)
- UDEFB (fonctions élémentaires dérivées et blocs fonctions élémentaires dérivés), (voir *UDEFB, p. 251*)

EFB

EFB est le terme générique pour :
- Fonction élémentaire (voir *Fonction élémentaire, p. 248*)
- Bloc fonction élémentaire (voir *Bloc fonction élémentaire, p. 249*)

Les EFB sont des fonctions et de blocs fonctions que Concept met à votre disposition sous forme de bibliothèques. La logique des EFB est établie en langage de programmation C et ne peut pas être modifiée dans l'éditeur FBD.

Note : Les EFB AND_BOOL, NOT_BOOL, OR_BOOL, R_TRIG et F_TRIG ne sont pas disponibles dans LD. Leur fonctionnalité est réalisée à l'aide de contacts. La fonction MOVE ne peut pas être utilisée avec le type de données BOOL.

Fonction élémentaire

Les fonctions n'ont pas d'état interne. Pour des valeurs égales aux entrées, la valeur à la sortie est la même pour toutes les exécutions de la fonction. Par exemple, l'addition de deux valeurs livre le même résultat à chaque exécution de la fonction.

Une Fonction élémentaire est représentée graphiquement comme un cadre avec des entrées et une sortie. Les entrées sont toujours représentées à la gauche et les sorties toujours à la droite du cadre. Le nom de la fonction, c'est-à-dire le type de fonction, est affiché au centre du cadre. Le numéro de fonction est affiché au-dessus du cadre.

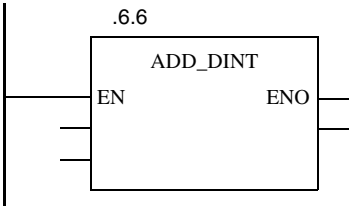
Le numéro de fonction ne peut pas être modifié et présente toujours la structure .n.m.

.n = numéro courant de la section

.n = numéro courant de la fonction

Les fonctions ne sont exécutées que lorsque l'entrée EN=1 ou lorsque l'entrée EN est désactivée (voir également *EN et ENO, p. 252*).

Fonction élémentaire



Bloc fonction élémentaire

Les blocs fonctions ont des états internes. Pour des valeurs égales aux entrées, la valeur à la sortie est toujours différente à chaque exécution du bloc fonction. Par exemple, pour un compteur, la valeur à la sortie augmente.

Un bloc fonction est représenté graphiquement comme un cadre avec des entrées et des sorties. Les entrées sont toujours représentées à la gauche et les sorties toujours à la droite du cadre. Le nom du bloc fonction, c'est-à-dire le type bloc fonction, est affiché au centre du cadre. Le nom d'instance est affiché au-dessus du cadre. Le nom d'instance permet d'identifier précisément le bloc fonction dans un projet.

Le nom d'instance est généré automatiquement et présente la structure suivante : FBI_n_m

FBI = Instance de bloc fonction

n = Numéro de section (numéro courant)

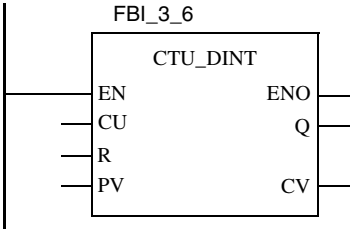
m = Numéro de l'objet FFB dans la section (numéro courant)

Le nom d'instance peut être édité dans la boîte de dialogue Propriétés du bloc fonction. Le nom d'instance doit être précis dans tout le projet sans égard aux majuscules et minuscules. Si le nom entré existe déjà, vous en êtes averti et vous devez choisir un autre nom. Le nom d'instance doit obligatoirement satisfaire aux conventions de nom CEI. Si cela n'est pas le cas, un message d'erreur apparaît.

Note : Selon CEI1131-3 seules des lettres sont autorisées comme premier caractère des noms d'instances. Si vous souhaitez cependant utiliser également des chiffres comme premiers caractères, vous pouvez le faire avec la commande **Options** → **Environnement** → **Extensions CEI** → **Chiffres en tête de qualificatif autorisés** Libérer.

Les blocs fonctions ne sont exécutés que lorsque l'entrée EN=1 ou lorsque l'entrée EN est désactivée (voir également *EN et ENO, p. 252*).

Bloc fonction élémentaire



DFB

Les blocs fonctions dérivés sont des blocs fonctions que vous avez définis dans Concept-DFB.

Dans les DFB, on ne distingue pas la fonction du bloc fonction. Indépendamment de leur structure interne, elles sont toujours traitées comme des blocs fonctions.

Un DFB est représenté graphiquement comme un cadre avec deux lignes verticales et des entrées et des sorties. Les entrées sont toujours représentées à la gauche et les sorties toujours à la droite du cadre. Le nom du DFB est affiché au centre du cadre. Le nom d'instance est affiché au-dessus du cadre. Le nom d'instance permet d'identifier précisément le bloc fonction dans un projet.

Le nom d'instance est généré automatiquement et présente la structure suivante : FBI_n_m

FBI = Instance de bloc fonction

n = Numéro de section (numéro courant)

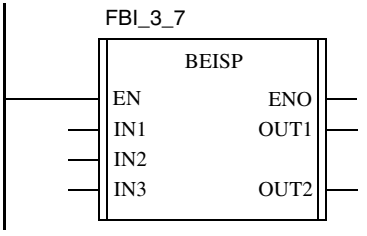
m = Numéro de l'objet FFB dans la section (numéro courant)

Le nom d'instance peut être édité dans la boîte de dialogue Propriétés du DFB. Le nom d'instance doit être précis dans tout le projet sans égard aux majuscules et minuscules. Si le nom entré existe déjà, vous en êtes averti et vous devez choisir un autre nom. Le nom d'instance doit obligatoirement satisfaire aux conventions de nom CEI. Si cela n'est pas le cas, un message d'erreur apparaît.

Note : Selon CEI1131-3 seules des lettres sont autorisées comme premier caractère des noms d'instances. Si vous souhaitez cependant utiliser également des chiffres comme premiers caractères, vous pouvez le faire avec la commande **Options** → **Environnement** → **Extensions CEI** → **Chiffres en tête de qualificatif autorisés** Libérer.

Les blocs fonctions dérivés ne sont exécutés que lorsque l'entrée EN=1 ou lorsque l'entrée EN est désactivée (voir également *EN et ENO, p. 252*).

Bloc fonction dérivé



UDEFB

UDEFB est le terme générique pour :
- Fonction élémentaire définie par l'utilisateur
- Bloc fonction élémentaire défini par l'utilisateur

Les UDEFB sont des fonctions et blocs fonctions que vous avez programmés avec Concept-EFB en langage de programmation C++ et que vous mettez sous forme de bibliothèques à la disposition de Concept.

Il n'y a pas de différence fonctionnelle entre les UDEFB et les EFB dans Concept.

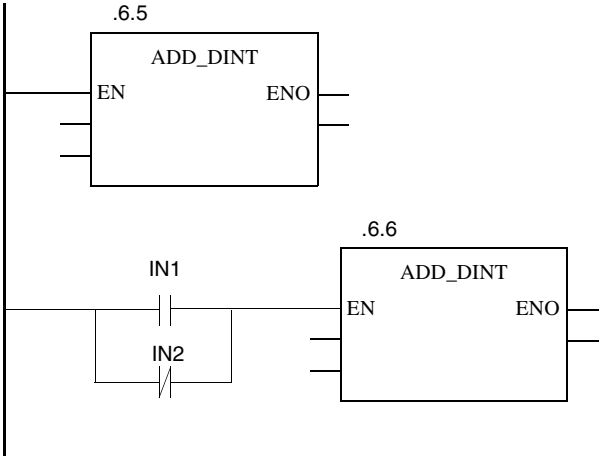
Traitement de FFB

Les FFB ne sont traités que lorsqu'au moins une entrée booléenne est connectée à la barre d'alimentation gauche. Au cas où le FFB ne possède pas d'entrée booléenne, l'entrée EN du FFB doit être utilisée. Si le FFB doit être exécuté de façon conditionnell, l'entrée booléenne peut être préalablement reliée par des contacts ou d'autres FFB.

Note : Si l'entrée EN n'est pas connectée à la barre gauche d'alimentation, elle doit être désactivée dans la boîte de dialogue Propriétés, car sinon le FFB ne sera jamais traité.

Note : Tout FFB sans liaison booléenne à la barre gauche d'alimentation provoque un message d'erreur lors du chargement dans l'API.

Liaison d'un FFB à la barre gauche d'alimentation :



EN et ENO

Pour tous les FFB, une entrée EN et une sortie ENO peuvent être configurées.

La configuration d'EN et d'ENO est activée ou désactivée dans la boîte de dialogue des Propriétés FFB. La boîte de dialogue est appelée par la commande **Objets** → **Caractéristiques...** ou en cliquant deux fois sur le FFB.

Au cas où la valeur d'EN est égale à "0", lorsque le FFB est appelé, les algorithmes définis par le FFB ne seront pas exécutés et les sorties conserveront leur valeur précédente. Dans ce cas, la valeur d'ENO sera automatiquement placée à "0"

Au cas où la valeur d'EN est égale à "1", lorsque le FFB est appelé, les algorithmes définis par le FFB sont exécutés. Après l'exécution exempte d'erreur de ces algorithmes, la valeur d'ENO est automatiquement placée à "1". Si une erreur se produit durant l'exécution de ces algorithmes, ENO est placée à "0".

Note : Si l'entrée EN n'est pas connectée à la barre gauche d'alimentation, elle doit être désactivée dans la boîte de dialogue Propriétés, car sinon le FFB ne sera jamais traité.

Le comportement de sortie des FFB est indépendant du fait que les FFB soient appelés sans EN/ENO ou avec EN=1 .

Liaison

Description

Les liaisons sont des liens entre des contacts, des bobines et des FFB.

Il est possible d'établir plusieurs liaisons avec un contact, une bobine ou une sortie FFB. Les différents points de liaison sont marqués par un cercle plein.

Note : Les contacts, bobines et entrées de FFB non liés comportent par défaut la valeur "0".

Types de données

Les types de données des entrées / sorties à relier doivent correspondre.

Editer des liaisons

Les liaisons peuvent être éditées en mode sélection. Les recouvrements avec d'autres objets sont autorisés.

Configurer des boucles

Aucune boucle ne peut être configurée avec des liaisons car, dans ce cas, l'ordre d'exécution dans la section ne peut pas être déterminé de manière unique. Les boucles doivent être définies avec des paramètres réels (voir *Configuration de boucles, p. 228*).

Liaisons horizontales

Lors de leur mise en place, les contacts et les bobines sont reliés automatiquement à une bobine / un contact voisin(e) non lié(e) qui se trouve à la même position verticale. Une liaison à la barre d'alimentation n'est établie que lorsque le contact est placé à proximité (voir également le sous-chapitre *Définir une liaison des contacts, p. 1157* dans le chapitre *Fichier INI Concept*). Si une bobine ou un contact est placé(e) sur une liaison horizontale existante, celle-ci est alors coupée automatiquement et le contact / la bobine est inséré(e). Lors de la mise en place, les paramètres actuels peuvent chevaucher un autre objet, mais ne peuvent pas franchir les limites du cadre de la section. Si une liaison est établie avec un autre objet via une connexion, cette liaison est vérifiée. Si cette liaison n'est pas autorisée, vous obtenez un message d'erreur et la liaison n'est pas générée.

Après la mise en place des objets, le système crée automatiquement les liaisons horizontales avec les objets directement voisins.

Liaisons verticales

La "liaison verticale" est un cas particulier de liaison. La liaison verticale sert de OU logique. Pour ce type de liaison OU, 32 entrées (contacts) et 64 sorties (bobines, liaisons) sont possibles.

Paramètres réels

Paramètres réels possibles

Durant l'exécution du programme, les valeurs sont transmises, par le biais de paramètres réels, du processus ou d'autres paramètres réels au FFB, et renvoyées à nouveau à la sortie après le traitement.

Tableau des paramètres réels possibles

Elément	Paramètre réel
Contacts	• Adresse directe (voir *Adresses directes, p. 53*) • Variables affectées (voir *Variables :, p. 50*) • Variable non-affectée (voir *Variables :, p. 50*)
Bobines	• Adresse directe (voir *Adresses directes, p. 53*) • Variables affectées (voir *Variables :, p. 50*) • Variable non-affectée (voir *Variables :, p. 50*)
Entrées de FFB	• Adresse directe (voir *Adresses directes, p. 53*) • Variables affectées (voir *Variables :, p. 50*) • Variable non-affectée (voir *Variables :, p. 50*) • Constantes (voir *Constantes, p. 51*) • Libellés (voir *Littéraux (valeurs), p. 52*)
Sorties de FFB	• Adresse directe (voir *Adresses directes, p. 53*) • Variables affectées (voir *Variables :, p. 50*) • Variable non-affectée (voir *Variables :, p. 50*)

Adresses directes

L'indication/l'affichage des adresses directes peut s'effectuer dans différents formats. La définition du format d'affichage s'effectue dans le dialogue **Options** → **Environnement** → **Commun...**. La définition du format d'affichage n'a aucun impact sur le format d'entrée, ceci signifie que l'entrée d'adresses directes peut s'effectuer dans un format quelconque.

Les formats d'adresses suivants sont disponibles :

- **Format standard (400001)**
  L'adresse à cinq positions intervient directement après le premier chiffre (de la Référence).
- **Format à séparateur (4 : 00001)**
  Le premier chiffre (la référence) est séparé par un point double (:) de l'adresse à cinq positions.
- **Format compact (4 : 1)**
  Le premier chiffre (la référence) est séparé par un point double (:) de l'adresse suivante, les zéros en tête n'étant pas indiqués dans l'adresse .
- **Format CEI (QW1)**
  Un qualificatif est placé en première position, conforme CEI, suivi de l'adresse à cinq positions.
  - %0x12345 = %Q12345
  - %1x12345 = %I12345
  - %3x12345 = %IW12345
  - %4x12345 = %QW12345

Type de données

Pour les contacts et les bobines, le type de données du paramètre réel doit être de type BOOL. Pour les entrées/sorties, le type des données du paramètre réel doit correspondre au type des données de l'entrée/la sortie. La seule exception concerne les entrées/sortie génériques de FFB dont le type de données est déterminé par le paramètre formel. On choisira un type de données adapté pour le bloc fonction, si tous les paramètres réels sont constitués de libellés.

Valeurs initiales

Les FFB utilisant aux entrées des paramètres réels ainsi que les bobines auxquels aucune valeur n'a encore été affectée, fonctionnent avec les valeurs initiales de ces paramètres réels.

Entrées non reliées

Note : De façon standard, les contacts, bobines et entrées/sorties de FFB non reliés sont occupés par la valeur "0".

Objet texte

Introduction

Dans le plan de contacts LD, les textes peuvent être placés sous forme d'objets texte. La taille de ces objets texte est fonction de la longueur du texte. Selon la longueur du texte, la taille de l'objet peut être agrandie, dans les sens vertical et horizontal, d'unités de grille supplémentaires. Les objets texte ne doivent pas se chevaucher avec d'autres objets, le chevauchement avec des liaisons est toutefois possible.

Capacité de mémoire

Les objets texte n'occupent pas d'espace de mémoire sur l'API, étant donné que le texte n'est pas chargé sur l'API.

8.3 Tâches exécutées en langage à contacts LD

Introduction

Aperçu

Ce chapitre décrit ce que l'on peut faire avec le langage à contacts LD.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Placement de bobines, de contacts, de fonctions et de blocs fonctions	258
Ordre d'exécution	260
Configuration de boucles	262

Placement de bobines, de contacts, de fonctions et de blocs fonctions

Mise en place d'objets

Dans l'éditeur du schéma à contacts LD, la fenêtre est placée sur une grille logique. Lors de leur mise en place, les objets sont ajustés sur le quadrillage de cette grille (52 x 230 champs). A l'exception des liaisons verticales, des FFB et des zones de texte, tous les éléments ont besoin exactement d'un champ sur la grille. Les objets ne peuvent être placés qu'à l'intérieur d'un tel champ. Si un objet est placé entre deux champs, l'objet est automatiquement inséré dans le champ qui suit.

Lorsqu'un objet est placé hors du cadre de la section ou en cas de chevauchement avec un autre objet, un message d'erreur apparaît, et l'objet n'est pas placé.

Lors du placement, les contacts et les bobines sont automatiquement reliés à un contact / une bobine non relié(e) se trouvant à proximité immédiate à condition que le contact / la bobine se trouve à la même position verticale. Une liaison avec la barre d'alimentation n'est établie que lorsque le contact est placé 2 champs plus loin. Si des contacts ou bobines sont placé(e)s sur des contacts ou bobines déjà existant(e)s, ces derniers sont remplacés par ceux actuellement mis en place (valable uniquement pour les mêmes types, c'est-à-dire lorsque des bobines sont remplacées par des bobines et des contacts par des contacts). Si une bobine ou un contact est placé(e) sur une liaison horizontale existante, celle-ci est coupée automatiquement et le contact / la bobine est inséré(e).

Lors de la mise en place, les paramètres réels peuvent chevaucher un autre objet, mais ne peuvent pas franchir les limites du cadre de la section. Si une liaison est établie avec un autre objet via une connexion, cette liaison est vérifiée. Si cette liaison n'est pas autorisée, vous obtenez un message d'erreur et la liaison n'est pas générée. Lors de la génération de liaisons, des chevauchements et croisements avec d'autres liaisons et objets sont permis.

Lorsqu'un FFB est sélectionné, le commentaire associé est affiché dans la première colonne de la ligne d'état. Lorsqu'un paramètre réel est sélectionné, son nom et, le cas échéant, son adresse directe et son commentaire apparaissent dans la première colonne de la ligne d'état.

Saut de ligne automatique

Si vous utilisez le clavier, vous avez la possibilité de définir dans le fichier CONCEPT.INI (voir *Définir le nombre de colonnes / champs, p. 1157*) le nombre de colonnes / champs autorisé avant que le système n'ajoute un saut de ligne automatique en cas d'édition avec le clavier lorsque vous atteignez la dernière colonne / le dernier champ. L'objet qui suit est alors inséré dans la deuxième colonne / le deuxième champ et relié au dernier objet de la ligne précédente. Ainsi, les objets sont créés au sein d'une même voie de courant.

Sélectionner des FFB

La commande **Objets** → **Sélection FFB...** vous permet d'ouvrir un dialogue pour sélectionner des FFB. Ce dialogue est "modeless", c.-à-d. qu'il ne se referme pas automatiquement après le placement d'un FFB, mais reste ouvert jusqu'à ce que vous le fermiez. Si vous avez ouvert plusieurs sections LD et appelé le dialogue, seul un dialogue disponible pour toutes les sections s'ouvre. Pour toutes les autres sections (pas l'éditeur LD), la boîte de dialogue n'est pas disponible. Lorsque les sections LD sont transformées en symboles (réduction de la fenêtre), la boîte de dialogue se ferme. Lorsque l'un des symboles de section LD est à nouveau appelé, la boîte de dialogue s'ouvre à nouveau automatiquement.

Au premier démarrage de Concept, l'affichage des FFB s'oriente sur les bibliothèques. Ceci signifie que pour sélectionner un FFB, il faut d'abord sélectionner la bibliothèque correspondante à l'aide du bouton de commande **Bibliothèque**. Vous pouvez ensuite sélectionner le groupe correspondant dans la zone de liste **Groupe**. Vous pouvez alors sélectionner le FFB souhaité dans la zone de liste Type FFB.

Si vous ne savez pas dans quelle bibliothèque / dans quel groupe le FFB souhaité se trouve, vous pouvez utiliser le bouton de commande **Orienté FFB** pour appeler un dialogue orienté sur les FFB. Ce dialogue présente une liste triée par ordre alphabétique de tous les FFB de toutes les bibliothèques et de tous les groupes.

Après chaque démarrage ultérieur du projet, la vue que vous avez sélectionnée apparaît.

Après avoir sélectionné le FFB, vous devez choisir sa position dans la section. Le symbole curseur se transforme en petit FFB, la croix indiquant la position (coin supérieur gauche du FFB) à laquelle est placé le FFB. Cliquez sur le bouton gauche de la souris pour placer le FFB.

Modifier type FFB

La commande **Objets** → **Remplacer FFB...** vous permet de remplacer des FFB déjà placés dans la section par des FFB d'un autre type (par ex. un ET par un OU). Cependant, les variables attribuées au FFB restent préservées lorsque le type de données et la position des entrées / sorties de "l'ancien" et du nouveau FFB correspondent.

Note : Il est impossible de remplacer des FFB avec des entrées / sorties de données du type ANY (FFB génériques).

Modifier contact / bobine

Les contacts et bobines déjà placés peuvent être remplacés facilement. Pour ce faire, sélectionnez le nouvel élément et cliquez sur l'élément à remplacer.

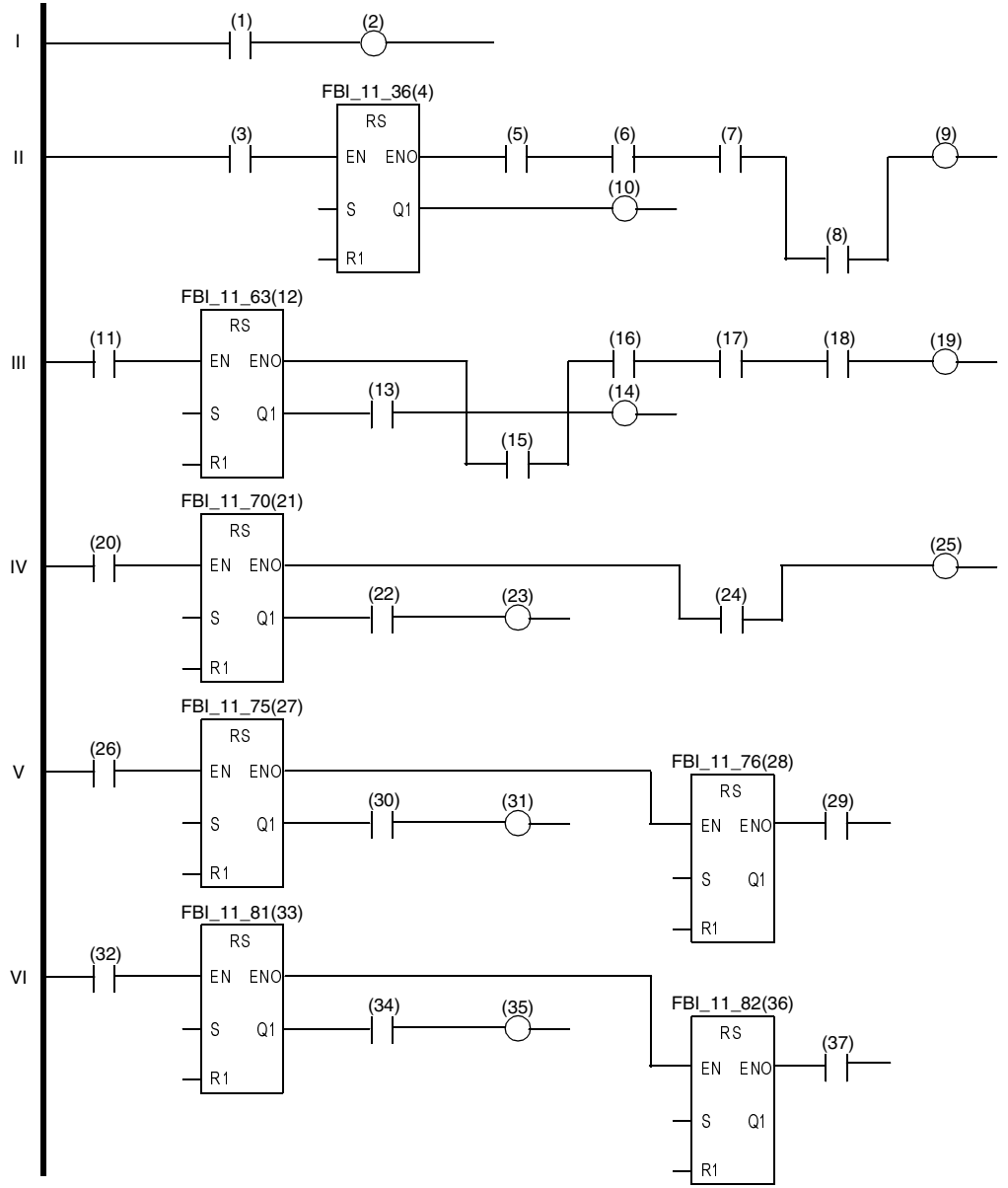
Ordre d'exécution

Description

La section suivante décrit l'ordre d'exécution des sections LD :

- L'ordre d'exécution des réseaux, qui sont reliés uniquement par la barre d'alimentation gauche, est déterminé par la position graphique où ces réseaux sont reliés à la barre d'alimentation gauche.
  L'exécution des différents réseaux s'effectue de haut en bas.
  Voir l'exemple ci-dessous, Réseaux I-VI.
- L'ordre d'exécution des différents objets (contacts, bobines, FFB) au sein d'un réseau est déterminé par le flux de données. Cela signifie que les bobines et FFB dont les entrées ont déjà reçu des affectations de valeurs sont traités en premier.
- Les voies de courant, qui commencent au niveau des sorties (broches) des FFB, sont traitées en fonction de la position graphique verticale de leur premier objet (de haut en bas).
  Voir l'exemple ci-dessous, Réseau III :
  Le traitement des objets du FFB (FBI_11_63) débute avec la voie de courant dont le premier objet occupe la position verticale la plus haute (13), puis suit la voie de courant (13)->(14).
  Lorsque la voie de courant (13)->(14) a été traitée, le processus passe à la voie suivante (15)->(19).
- Lorsque les premiers objets de deux voies de courant commençant au niveau des sorties (broches) des FFB se situent au même niveau, la voie de courant dont l'objet est le plus à gauche est traitée en premier.
  Voir l'exemple ci-dessous, Réseau IV : (22)->(23), puis (24)->(25).
- La position d'un FFB est déterminée en fonction de son coin supérieur gauche.
  Voir exemple ci-dessous.
  Réseau V : Coin supérieur gauche du FFB (FBI_11_76) au-dessus du contact (30). Ordre : (28)->(29), puis (30)->(31).
  Réseau VI : Coin supérieur gauche du FFB (FBI_11_82) à la même hauteur que le contact (34). Ordre : (34)->(35), puis (36)->(37).

Exemple Section LD

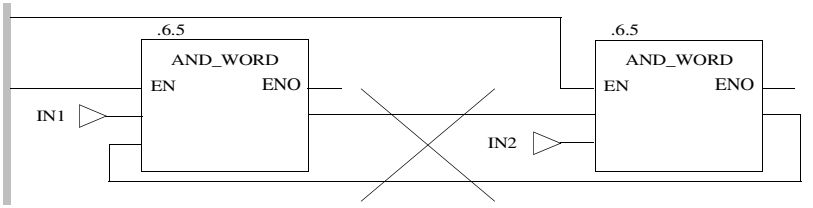


Configuration de boucles

Boucles non permises

La configuration de boucles exclusivement par le biais de liaisons n'est pas permise, étant donné que, dans ce cas, une détermination unique du flux de données n'est pas possible (la sortie d'un FFB est l'entrée du FFB suivant, et la sortie de celui-ci est à son tour l'entrée du premier).

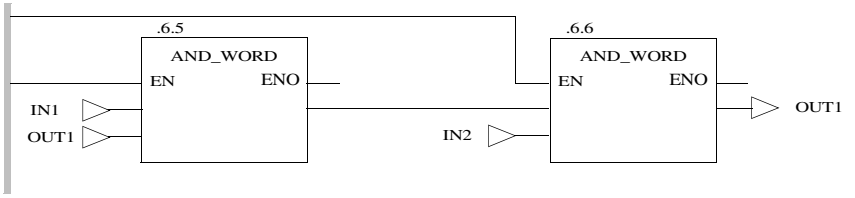
Boucles non permises par le biais de liaisons



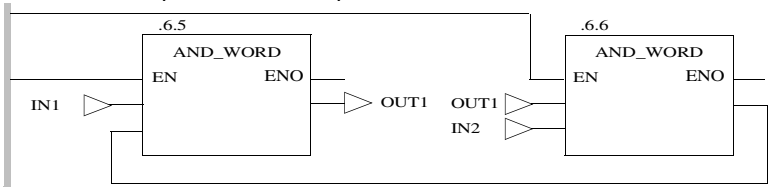
Résolution par le biais d'un paramètre réel

Une telle logique doit être résolue par le biais de paramètres réels, afin que le flux de données puisse être défini de façon unique .

Boucle résolue par le biais d'un paramètre réel : Variante 1



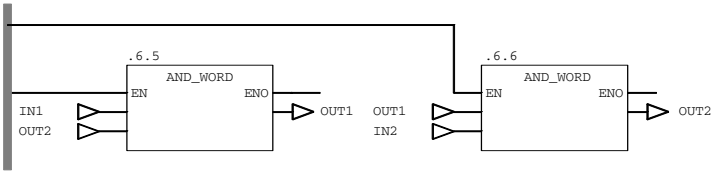
Boucle résolue par le biais d'un paramètre réel : Variante 2



Résolution par le biais de plusieurs paramètres réels

Les boucles résolues par le biais de plusieurs paramètres réels sont également permises.

Boucle par le biais de plusieurs paramètres réels



8.4 Génération de codes dans le cadre du langage à contacts LD

Options de génération de code

Introduction

Avec la commande **Projet** → **Options de génération de code**,vous pouvez définir les Options de génération de code.

Des données de diagnostic comprennent

Si vous activez la case à cocher **Inclure les données de diagnostic**, des informations pour le diagnostic de processus seront générées en plus lors de la génération de codes (par ex. Diagnostic de transition, codes de diagnostic pour des blocs fonctions avec diagnostic étendu, comme par exemple XACT, XLOCK, etc.). L'évaluation de ce diagnostic de processus peut s'effectuer avec MonitorPro ou FactoryLink, par exemple.

Code le plus rapide (vérification limitée)

Si vous activez la case à cocher **Code le plus rapide (vérification limitée)**, un code à temps optimisé sera généré. L'optimisation du temps d'exécution est obtenue de la façon suivante : l'arithmétique sur les entiers (par ex. "+" ou "-") est réalisée via des commandes simples de processeur, au lieu d'être réalisée par des appels de EFB.

Les commandes de processeur sont bien plus rapides que les appels de EFB, mais ils ne génèrent pas de messages d'erreur, comme Dépassement en arithmétique ou Débordement de tableau (d'Array). Vous ne devez utiliser cette option que si vous vous êtes assuré que le programme ne contient aucune erreur d'arithmétique.

Si **Code le plus rapide (vérification limitée)** a été sélectionné, l'addition IN + 1 est exécutée par la commande de processeur "add". Le code est plus rapide que lorsque l'EFB ADD_INT est appelé. Mais aucune erreur d'exécution n'est générée si la valeur "IN1" est égale à 32767 . Dans ce cas, "OUT1" déborderait de 32767 à -32768.

8.5 Fonctions en ligne dans le cadre du langage à contacts LD

Fonctions en-ligne

Introduction

Deux modes d'animation sont disponibles dans l'éditeur LD :
- Animation de variables et liaisons binaires
- Animation d'objets sélectionnés

Ces modes sont également disponibles lors de l'affichage d'un instance DFB (Bouton de commande **Détail...** dans la boîte de dialogue **Bloc fonction : xxx**).

Note : Lorsque la section animée est utilisée comme section de transition pour SFC et la transition (et donc la section de transition) n'est pas traitée, l'état suivant apparaît dans la section de transition animée : **BLOQUE**.

Animation de variables et liaisons binaires

L'animation de variables et liaisons binaires est activée à l'aide de la commande **En ligne → Animer valeurs binaires**.

Dans ce mode, l'état de signal actuel des variable binaires, des adresses directes des domaines 0x et 1x et des liaisons binaires est affiché dans la fenêtre de l'éditeur.

Signification des couleurs

Couleur	Signification
Contact, bobine, entrée/sortie, connexion rouge	Contact, bobine, entrée/sortie, connexion transmet la valeur 0
Barre d'alimentation de gauche, contact, bobine, entrée/sortie, connexion vert(e)	Barre d'alimentation de gauche, contact, bobine, entrée/sortie, connexion transmet la valeur 1
Variable avec ombre beige	Variable forcée
Variable avec ombre violette	Variable définie de façon cyclique
Le nom des variables multi-éléments (par ex. moteur) est affiché en couleur.	Affichage, dans l'éditeur, d'une variable multi-éléments (moteur, par ex.) comprenant un ou plusieurs éléments, objet d'un forçage ou d'un positionnement cyclique.
Le nom intégral de l'élément des variables multi-éléments (par ex. moteur.droite.act.) est affiché en couleur.	Affichage, dans l'éditeur, d'un élément de variable multi-éléments (Moteur droit Marche, par ex), objet d'un forçage ou d'un positionnement cyclique.
Le nom des variables multi-éléments (par ex. moteur.droit.act.) est affiché en couleur, mais pas le nom de l'élément.	Dans l'éditeur est affiché un élément d'une variable multi-éléments (par ex. moteur.droite.act.). Cet élément n'est ni forcé, ni défini de façon cyclique, mais un autre élément de cette variable multi-éléments est défini de façon cyclique ou est forcé.

Animation d'objets sélectionnés

L'animation d'objets sélectionnés est activée à l'aide de la commande **En ligne** → **Animer sélection** .

Dans ce mode, l'état de signal actuel des liaisons, variables, variables multi-éléments et libellés sélectionnés est affiché dans la fenêtre de l'éditeur.

Note : Au cas où vous souhaiteriez animer toutes les variables/liaisons de la section, vous pouvez utiliser **CTRL**+**A**, pour marquer la section entière, et ensuite, avec **En ligne** → **Animer sélection** (**CTRL**+**W**), animer toutes les variables et liaisons de la section.

Si vous sélectionnez une valeur numérique à une entrée/sortie, le nom de la variable, son adresse directe et son affectation des E/S (si existant) ainsi que son commentaire s'affichent dans la ligne d'état.

Note : Les objets sélectionnés restent également sélectionnés après une nouvelle sélection de Animer sélection , afin de sauvegarder ces objets pour la continuation de la lecture, ou permettre une modification aisée de la liste des objets.

Signification des couleurs

Couleur	**Signification**
Contact, bobine, connexion rouge	Contact, bobine, connexion transmet la valeur 0
Barre d'alimentation de gauche, contact, bobine, connexion vert(e)	Barre d'alimentation de gauche, contact, bobine, connexion transmet la valeur 1
Connexion jaune	La connexion transmet une variable multi-éléments (cliquer deux fois sur la connexion pour afficher les valeurs)
Variable avec fond jaune	Variable multi-éléments (cliquer deux fois sur la variable pour afficher les valeurs)
Valeur numérique au niveau de l'entrée/ de la sortie (fond jaune)	Valeur actuelle des variables
Valeur numérique sur connexion (fond jaune)	Valeur transmise actuellement par la connexion
Variable avec ombre beige	La variable est forcée
Variable avec ombre violette	La variable est définie de façon cyclique
Le nom des variables multi-éléments (par ex. moteur) est affiché en couleur.	Dans l'éditeur est affichée une variable multi-éléments (par ex. moteur) dans laquelle un ou plusieurs éléments sont forcés ou définis de façon cyclique.

Couleur	Signification
Le nom intégral de l'élément des variables multi-éléments (par ex. moteur.droite.act.) est affiché en couleur.	Dans l'éditeur est affiché un élément d'une variable multi-éléments (par ex. moteur.droite.act). Cet élément est forcé ou défini de façon cyclique.
Le nom des variables multi-éléments (par ex. moteur.droit.act.) est affiché en couleur, mais pas le nom de l'élément.	Dans l'éditeur est affiché un élément d'une variable multi-éléments (par ex. moteur.droite.act.). Cet élément n'est ni forcé, ni défini de façon cyclique, mais un autre élément de cette variable multi-éléments est défini de façon cyclique ou est forcé.

Signification des couleurs

12 jeux de couleurs différents sont disponibles à des fins d'animation. Vous trouverez une vue d'ensemble des jeux de couleurs et de leur signification dans l'aide en ligne (astuce: recherchez dans l'aide en ligne en entrant "Couleurs" dans l'index).

8.6 Elaboration d'un programme avec le langage à contacts LD

Etablissement d'un programme dans le plan de contact LD

Introduction

La description suivante donne un exemple pour l'établissement d'un programme dans le plan de contacts (LD). L'établissement d'un programme dans le plan de contact LD s'effectue en deux étapes principales :

Etape	Action
1	Génération d'une section (voir *Génération d'une section, p. 268*)
2	Elaboration de la logique (voir *Elaboration de la logique, p. 269*)

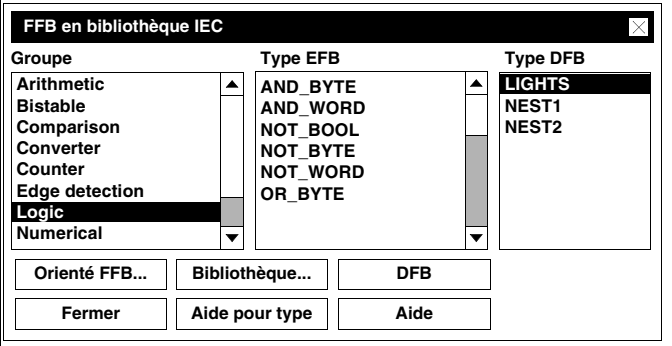
Génération d'une section

Pour générer une section, effectuez les opérations suivantes :

Etape	Action
1	Générez avec la commande **Fichier** → **Nouvelle section...** une nouvelle section et attribuez-lui un nom de la section. **Note :** Le nom de la section (au max. 32 caractères) doit être unique dans tout le projet, aucune différence n'étant faite entre l'écriture en lettres majuscules et en minuscules. Si le nom de la section entré existe déjà, vous recevez un avertissement et devez sélectionner un autre nom. Le nom de la section doit correspondre aux conventions de nom CEI, sinon un message d'erreur s'affiche. **Note :** Selon CEI 1131-3, uniquement des lettres sont permises comme premier caractère de noms. Si vous souhaitez cependant utiliser également des chiffres comme premiers caractères, vous pouvez le faire avec la commande **Options** → **Environnement** → **Extensions CEI** → **Chiffres en tête de qualificatif autorisés** Libérer.

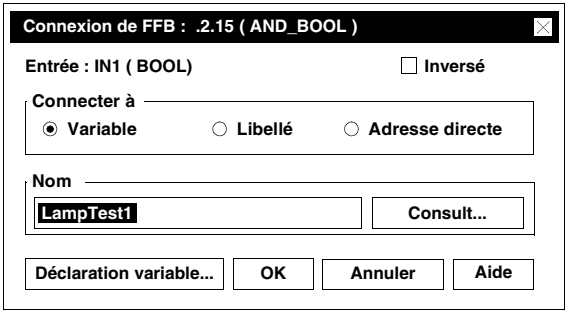
Elaboration de la logique

Pour élaborer une logique, effectuez les opérations suivantes :

Etape	Action
1	Pour insérer un contact ou une bobine dans la section, ouvrez le menu principal **Objets** et sélectionnez le contact ou la bobine souhaité(e). Vous pouvez également sélectionner le contact ou la bobine par la barre à outils. Placez les contacts ou les bobines dans la section.
2	Pour insérer un FFB dans la section, sélectionnez la commande **Objets** → **Sélection FFB...**. **Réaction :** La boîte de dialogue **FFB de bibliothèque** s'ouvre. 
3	Dans cette boîte de dialogue, vous pouvez employer le bouton de commande **Bibliothèque...** pour sélectionner une bibliothèque et y choisir un FFB. Mais vous pouvez également utiliser le bouton de commande **DFB** pour afficher les DFB que vous avez générés et en sélectionner un.
4	Placez uniquement le DFB sélectionné dans la section.
5	Lorsque vous avez placé tous les DFB sélectionnés, fermez la boîte de dialogue avec **Fermer**.
6	Avec **Objets** → **Mode sélection**, activez le mode sélection et déplacez les contacts, les bobines et les FFB sur la position souhaitée.
7	Avec **Objets** → **Liaison**, activez le mode de liaison et établissez les liaisons entre les contacts, les bobines et les FFB. Etablissez la liaison entre les contacts, les FFB et la barre gauche d'alimentation.
8	A présent, activez à nouveau le mode sélection avec **Objets** → **Mode sélection**, et cliquez deux fois sur un contact ou une bobine. **Réaction :** La boîte de dialogue **Propriétés : Objet LD** s'ouvre, et vous pouvez y affecter un paramètre réel au contact/à la bobine.

Etape	Action

9	Selon la logique du programme, vous pouvez affecter au contact/à la bobine :
	• **Variable**
	  • **Variable affectée**
	  Par une variable localisée , vous pouvez affecter un signal d'entrée/de sortie de matériel à l'entrée/la sortie.
	  Le nom de la variable est affiché sur la fenêtre de l'éditeur à l'entrée/la sortie.
	  • **Variable non-affectée**
	  Vous pouvez utiliser la variable non localisée affectée à l'entrée/la sortie comme mémento, c'est-à-dire pour la résolution de boucles ou pour la transmission de valeur entre différentes sections.
	  Le nom de la variable est affiché sur la fenêtre de l'éditeur à l'entrée/la sortie.
	• **Adresse directe**
	  Par une adresse, vous pouvez affecter un signal d'entrée/de sortie de matériel à l'entrée/la sortie.
	  L'adresse est affichée sur la fenêtre de l'éditeur à l'entrée/la sortie.
	Note : Exemple d'appel de variables multi-éléments, voir *Appel de types de données dérivés, p. 603*.
	Note : De façon standard, les entrées FFB non reliées sont occupées par "0".

10	Pour relier les entrées/sorties FFB aux paramètres réels, cliquez deux fois sur une des entrées/sorties non reliées. **Réaction :** La boîte de dialogue Connexion de FFB s'ouvre, dans laquelle vous pouvez affecter un paramètre réel à l'entrée/la sortie. 

Etape	Action

11	Selon la logique du programme, vous pouvez affecter à l'entrée/la sortie :
	• **Variable**
	  • Variable affectée
	  Par une variable localisée , vous pouvez affecter un signal d'entrée/de sortie de matériel à l'entrée/la sortie.
	  Le nom de la variable est affiché sur la fenêtre de l'éditeur à l'entrée/la sortie.
	  • Variable non-affectée
	  Vous pouvez utiliser la variable non localisée affectée à l'entrée/la sortie comme mémento, c'est-à-dire pour la résolution de boucles ou pour la transmission de valeur entre différentes sections.
	  Le nom de la variable est affiché sur la fenêtre de l'éditeur à l'entrée/la sortie.
	  • Constante
	  Vous pouvez affecter une constante à l'entrée. Vous pouvez transmettre la constante à d'autres sections. Vous déterminez la valeur des constantes dans l'éditeur des variables.
	  Le nom de la constante est affiché sur la fenêtre de l'éditeur à l'entrée.
	• **Libellé**
	  Vous pouvez affecter un libellé à l'entrée, c'est-à-dire que vous pouvez affecter directement une valeur à l'entrée/la sortie.
	  La valeur est affichée sur la fenêtre de l'éditeur à l'entrée.
	• **Adresse directe**
	  Par une adresse, vous pouvez affecter un signal d'entrée/de sortie de matériel à l'entrée/la sortie.
	  L'adresse est affichée sur la fenêtre de l'éditeur à l'entrée/la sortie.
	Note : Exemple d'appel de variables multi-éléments, voir *Appel de types de données dérivés, p. 603*.
	Note : De façon standard, les entrées FFB non reliées sont occupées par "0".

12	Sauvegardez à présent la section LD avec la commande **Fichier** → **Enregistrer projet**.

Index

=>Affectation, 379, 451
'Variable d'étape, 280
'Variable SFCSTEP_STATE', 280
'Variable SFCSTEP_TIMES', 279

A

Abonnés réseau Modbus Plus, 118
Action, 281
actions
    traiter, 301
Activer les boîtes de dialogue, 117
Adresses directes, 49
Affectation
    =>, 451
Affectation des E/S, 111
Affectation des E/S..., 67
Affichage de diagnostic, 707
Affichage des événements
    configuration INI, 1165
Aide, 849
Aide contextuelle, 849
Aide en ligne, 849
Animation, 622, 775, 777
    FBD, 231
    IL, 395
    IL/ST, 392
    LD, 265
    Remarques générales, 703
    Section, 703
Animation, 622, 775, 777
    Section CEI, 704
    Section LL984, 706
    SFC, 314, 316
Annuler
    LL984, 472
API
    Simuler, 773, 775, 777
Appel
    FFB, 383
Appel Bloc
    IL, 375
Appel composant
    ST, 447
Appeler
    Projet, 846
    projet, 846
Application protégée, 22
Apprendre temps de surveillance
    SFC, 320
Archiver
    DFB, 768
    EFB, 768
    projet, 768
ARRAY
    contrôle de plage, 607
Atrium
    Optimiser la mémoire, 205
Automate
    Etat, 838
Autorisations d'accès, 790, 799, 801

B

Backplane Expander Config
Configure, 127
Barre d'outils, 867
Barre d'outils, 857, 858
Barre d'état, 838
Barre d'outils, 855, 857, 858, 859, 860, 862, 863, 864, 866
Barre d'outils service, 855
Bibliothèques, 12
Bloc fonction
FBD, 216
LD, 249
Bloc fonction dérivé
FBD, 217
LD, 250
Bloc fonction élémentaire
FBD, 216
LD, 249
Bloc fonction élémentaire défini par l'utilisateur
FBD, 218
LD, 251
Blocs de sections d'interruption, 1195
Bobine
modifier, LD, 259
remplacer, LD, 259
Bobine – Front négatif
LD, 245
Bobine – Front positif
LD, 245
Bobine – Initialiser
LD, 246
Bobine – Inversée
LD, 245
Bobine – Réinitialiser
LD, 246
Bobines
LD, 244
Boîtes de dialogue, 843
boucle de chaîne, 288
Boucle de cycle
constant, 671
unique, 672
Boucle de cycle constant, 671
Boucles
FBD, 228
LD, 262
Bridge Modbus Plus, 664

C

caractères de définition, 304
CEI
Première mise en service Momentum, 1078, 1116, 1136
Chaîne
commander, 316
chaînes simples, 287
Chargement, 692
Chargement d'un projet, 689
Remarques générales, 690
Chargement des données de référence, 626
Chargement des modifications, 695
Charger le micrologiciel, 1143
Charger un projet, 689
Chemin d'accès aux DFB globaux
paramètres dans le fichier INI, 1153
Chemin d'accès aux fichiers d'aide
paramètres dans le fichier INI, 1153
Chemin de routage Modbus Plus
connexion automatique, 1198, 1202
Chiffrer un fichier journal, 22
Chiffrer un journal, 789
Clés
Editeur de type de données, 593
Type de données dérivé, 593
Combinaison de touches, 855
Commande décentralisée avec DIO, 944
Commande décentralisée avec RIO, 924
Commande décentralisée avec RIO (Série 800), 932
Commande en ligne, 676, 680
Commande et installation des API, 669
Commande IL
Appel fonction, 383
Commentaires, 343
Déclaration, 341
Inverser, 358
Lancement de bloc fonction, 376

Commande IL
Lancement de DFB, 376
OU logique, 354
OU logique exclusif, 356
VAR...END_VAR, 341
Commande INTERBUS, 950
Commande INTERBUS avec l'Atrium, 988
Commande Profibus DP, 964
Commande ST
, 423, 424, 425
-, 413, 419
&, 426
(), 410
*, 415
**, 412
+, 418
/, 416
=, 422
>, 420
>=, 421
Addition, 418
Affectation, 431
AND, 426
CASE...OF...END_CASE, 438
Commentaire, 446
Complémentation, 414
Déclaration, 433
Différents, 423
Division, 416
Egaux, 422
ELSE, 436
ET logique, 426
EXIT, 444
Exponentiation, 412
FOR...TO...BY...DO...END_FOR, 439
FUNCNAME, 411
IF...THEN...END_IF, 435
inférieur à, 424
inférieur ou égal à, 425
Instruction d'espacement, 445
Lancement de bloc fonction, 448
Lancement de fonction, 452
Mise entre parenthèses, 410
MOD, 417
Commande ST
Modulo, 417
Multiplication, 415
Négation, 413
NOT, 414
OR, 427
OU exclusif logique (XOR), 428
OU logique, 427
REPEAT...UNTIL...END_REPEAT, 443
Soustraction, 419
supérieur à, 420
supérieur ou égal à, 421
VAR...END_VAR, 433
WHILE...DO...END_WHILE, 442
XOR, 428
commande ST
ELSIF...THEN, 437
Commandes, 841
Comment utiliser le scrutateur d'E/S / Ethernet
Scrutateur d'E/S / Ethernet, 142
Commentaire
Editeur de types de données, 600
Type de données dérivé, 600
Communication, 21
Communication de diffusion des E/S, 69
Communication Modbus, 68
Compact
Optimiser mémoire, 185
Comportement au démarrage de sorties TOR, 53
Comportement de démarrage
variables, 51
Concept M
Contenu de paquet du matériel, 7
Concept ModConnect, 1027
Intégration de nouveaux modules, 1032
Retrait de modules, 1033
Utilisation de modules tiers dans Concept, 1035
Concept S
Contenu de paquet du matériel, 7
Concept XL
Contenu de paquet du matériel, 7

CONCEPT.INI, 1147, 1149
chemin d'accès aux DFB globaux, 1153
chemin d'accès aux fichiers d'aide, 1153
configuration de l'édition en ligne, 1158
configuration de messages d'avertissement, 1159
configuration du dépôt de variables, 1152
configuration du format d'adresse, 1159
configuration du format d'adresse du registre, 1152
définition de nom de projet, 1152
emplacement des DFB globaux lors de la lecture, 1153
exclusion des DFB globaux/locaux de l'enregistrement en ligne, 1153
généralités, 1150
lecture de DFB globaux, 1153
paramètres d'impression, 1151
paramètres de l'éditeur de données, 1161
paramètres de sécurité, 1160
paramètres du menu Tools, 1162
paramètres pour la section LD, 1157
représentation de données internes, 1156
Concept-DFB, 491, 535
Concept-Security, 787, 790, 799, 801, 802
Concept-SIM, 775, 777
Condition préalable requise pour une configuration inconditionnelle, 98
Configurateur
Scrutateur E/S / Ethernet, 138
Configurateur API
Symboles, 865
Configuration, 87
Allgemeines, 89
En option, 115
inconditionnelle, 97
Configuration API, 87
Export, 754, 755
Import, 754, 755
Remarques générales, 90
Configuration automate, 64, 65
Configuration Compact
Extension RTU, 137
Configuration d'interface dans Windows 98/ 2000/XP
environnement Modbus, 1062
Configuration de différents systèmes de réseau, 131
Configuration en mode HORS LIGNE et EN LIGNE
Remarques générales, 94
Configuration en mode local et connecté, 93
Configuration inconditionnelle, 97
Condition préalable, 98
Configuration minimale, 65
Configuration optionnelle, 115
Configuration réseau
TCP/IP, 1009
Configure
Backplane Expander Config, 127
Configurer, 65
DP Profibus, 133
Ethernet, 135
Extension RTU, 137
INTERBUS, 132
Configurer des systèmes de réseau, 131
Configurer Ethernet, 135
Configurer le système DP Profibus, 133
Configurer les systèmes de réseau, 117
Configurer système INTERBUS, 132
Conformité CEI, 887
Connecter API, 653
Connexion à l'API
automatique via des paramètres de ligne de commande, 1198
automatique via l'outil CCLaunch, 1202
Connexion automatique, 1198, 1202
Connexion de l'API
généralités, 654
Constantes, 49
Contact
modifier, LD, 259
remplacer, LD, 259
Contact de fermeture
LD, 242
Contact de rupture
LD, 242
Contacts
LD, 242, 243

Contenu de paquet du matériel API en
Concept S, M et XL, 7
Contrôle de plage
ARRAY, 607
Conversion
Programmes Modsoft, 1037
Conversion de Tableaux Editeur de
données, 614
Convertir
DFB, 1023
Macros, 1023
Projets, 1023
Créer
DFB, 513
Macro, 547
Programme, 61
Projet, 61
Symbole de projet, 846
symbole de projet, 846
Cycles uniques, 672

D

DDT, 589
Décalage de références
LL984, 471
déclaration de variables, 561
Déclarer
Transition, 307
déclarer
actions, 301
propriétés d'étapes, 299
déclarer des variables, 561
Définir le nombre de colonnes LD / de
champs
paramètres dans le fichier INI, 1157
Définir les couleurs
fichier INI, 1158
Définir une liaison des contacts LD
paramètres dans le fichier INI, 1157
définition cyclique de variables
éditeur de données de référence, 618
Définition de nom de projet
configuration de fichier INI, 1152
Définition de type de données
avancée (supérieure à 64 Ko), 589
Démarrage à chaud, 51
Démarrage à froid, 51
Dépôt de variables
configuration de fichier INI, 1152
dériviation simultanée, 292
Désactiver
section, 56
sections d'interruption, 56
Désignations pseudonymes
Etape, 309
Transition, 309
DFB, 491, 494
Aide sur contexte, 511
archiver, 768
Convertir, 1023
Création de variables globales, 507
Créer, 513
Documentation, 759
Effacer, 771
FBD, 217
global, 496
Lancement, 372, 376, 448
LD, 250
local, 496
Protéger, 802
DFB global, 496
DFB globaux
définir un chemin d'accès, 1153
déplacer, 1154
fichier INI, 1153
lire, 1155
DFB local, 496
Diagnostic
Diagnostic de transition, 323
Diagnostic de transition, 323
Diagnostic en ligne, 707
Différents, 654
Diffusion des E/S, 118, 976
divergence en OU, 289
Divers réglages API, 70
Documentation
DFB, 759
Macro, 759
Mise en page, 761
Mots-clés, 766

Documentation
Projet, 759
Sommaire, 760
Données Standby Hot CEI, 105
DOS-Loader
Première mise en service Atrium, 1133
Première mise en service Compact, 1093, 1130
Première mise en service dans le cas de l'utilisation de Modbus, 1089
Première mise en service dans le cas de l'utilisation de Modbus Plus, 1125
Première mise en service Momentum, 1096, 1099, 1136, 1139
Première mise en service Quantum, 1090, 1126
DTY, 581, 583, 584

E

E/S
forcer, 612
Echange de données entre abonnés au réseau Modbus Plus, 118
ED
Conversion de Tableaux Editeur de donnéesiteur de données, 614
définition cyclique de variables, 618
Editer
Transition, 307
éditer
propriétés d'étapes, 299
Editer la station E/S locale, 920, 951
Editeur de données
Paramètres dans le fichier INI, 1161
remarques générales, 612
Editeur de données de référence
Conversion de Tableaux Editeur de donnéesiteur de données, 614
remarques générales, 612
Remplacement de variables, 625
éditeur de données de référence
définition cyclique de variables, 618
modification d'états de signal d'une variable localisée, 616
Editeur de message ASCII
Comment utiliser, 640
Interface utilisateur, 640
Editeur de messages ASCII, 627, 629, 635
Code de contrôle, 633
Comment continuer après un avertissement, 645
Espaces, 634
Généralités, 630
Interface utilisateur, 639
Mode combinaison, 647
Mode direct, 647
Mode local, 647
Numéro de message, 642
Répétition, 637
Texte, 631
Texte de simulation, 644
Texte message, 643
Variables, 632
Vidage (tampon), 636
Editeur de type de données, 581
Clés, 593
Occupation de la mémoire, 601
Touches courtes, 871
Editeur de types de données, 583, 584
Commentaire, 600
Eléments, 592
Noms, 597
séparateurs, 598
Syntaxe, 591
Editeur de variables, 559
éditeur de variables
déclaration, 561
Export de variables localisées, 572
Recherche et insertion, 568
rechercher et remplacer, 564
Remarques générales, 560
Editeur des données de référence, 611
Editeur ED
Barre d'outils, 866
Editeurs, 15
Edition
LL984, 472
Edition Réseaux
LL984, 474

EFB
archiver, 768
FBD, 215
LD, 247
EFB de sections d'interruption, 1195
Effacer
DFB, 771
Macro, 771
Projet, 771
Effacer contenu de l'API, 674
Effacer des zones de mémoire sur l'API, 674
Egal, 654
Elaboration de programme
IL, 397
Eléments
Editeur de types de données, 592
Type de données dérivé, 592
Eléments de fenêtre, 838
Emplacement des DFB globaux lors de la lecture
paramètres dans le fichier INI, 1153
EN
FBD, 218
LD, 252
En ligne, 775, 777
fichier INI, 1158
SFC, 313
ENO
FBD, 218
LD, 252
Enregistrement en ligne
paramètres INI, 1166
Enregistrer en flash, 676
Environnement Modbus
configuration d'interface dans Windows 98/2000/XP, 1062
Environnement Modbus Plus
installation du SA85/PCI85, 1048, 1052
Erreur d'exécution
sections d'evénements d'E/S, 1193
Etablissement de la connexion matérielle
Préréglages Modbus, 1065
Préréglages Modbus Plus, 1060
Etablissement du programme
FBD, 233
LD, 268
Etape, 278
Désignations pseudonymes, 309
Etape d'attente, 278
Etape initiale, 278
Etat, 654
Etat API, 654
Etat de l'API, 668, 685
Ethernet, 666
Ethernet avec Momentum, 136
Ethernet avec Quantum, 135
Ethernet MMS
Spécifier les unités couplées, 117
Ethernet Symax
Spécifier les unités couplées, 117
Ethernet TCP/IP
Spécifier les unités couplées, 117
Exclusion des DFB globaux/locaux de l'enregistrement en ligne
paramètres dans le fichier INI, 1153
EXECLoader
Première mise en service Atrium, 1112
Première mise en service Compact, 1073, 1108
Première mise en service en cas d'utilisation de Modbus, 1067
Première mise en service en cas d'utilisation de Modbus Plus, 1103
Première mise en service Momentum, 1078, 1083, 1116, 1120
Première mise en service Quantum, 1068, 1104
Exemple de Atrium
Commande INTERBUS, 987
Exemple de Compact, 983
Exemple de configuration
Atrium-Commande INTERBUS, 987
Commande compacte, 983
Momentum-Bus décentralisé d'E/S, 997
Momentum-Système de bus Ethernet, 1007
Quantum-Commande décentralisée avec DIO, 939
Quantum-Commande décentralisée avec RIO, 919
Quantum-Commande décentralisée

avec RIO (série 800), 927
Quantum-Commande INTERBUS, 949
Quantum-Commande Profibus DP, 963
Quantum-Commande SY/MAX, 955
Quantum-Diffusion des E/S, 975
Exemple de configuration de périphériques
Atrium-Commande INTERBUS, 987
Commande Compact, 983
Quantum-Commande décentralisée avec, 919
Quantum-Commande décentralisée avec DIO, 939
Quantum-Commande décentralisée avec RIO (série 800), 927
Quantum-Commande INTERBUS, 949
Quantum-Commande Profibus DP, 963
Quantum-Commande SY/MAX, 955
Quantum-Diffusion des E/S, 975
Exemple de configuration du matériel
Momentum-Bus décentralisé d'E/S, 997
Exemple de configuration matérielle
Momentum-Système de bus Ethernet, 1007
Exemple de Momentum
Bus décentralisé d'E/S, 997
Système de bus Ethernet, 1007
Exemple de Quantum
Commande décentralisée avec DIO, 939
Commande décentralisée avec RIO (série 800), 927
Commande INTERBUS, 949
Commande Profibus DP, 963
CommandesY/MAX, 955
Diffusion des E/S, 975
Quantum-Commande décentralisée avec RIO, 919
Exigences
LL984, 468
Export, 713
Configuration API, 754, 755
Section, 717
Type de données dérivé, 721
Variable, 721
Export de variables localisées, 572
Exportation
remarques générales, 715
Expressions
ST, 402
Extenseur d'arrière-plan
Généralités, 128
Modifier la répartition E/S, 129
Traitement des erreurs, 130
Extension RTU
Configuration Compact, 137
Configurer, 137
Extensions de configurations, 117
Extraction d'informations de processus
Etat et mémoire, 683
Extraire informations de processus
Etat et mémoire, 683
Extraire les informations de processus
Remarques générales, 684

F

Factory Link, 751
FBD
animation, 231
Appeler macro, 556
Barre d'outils, 859
Bloc fonction, 216
Bloc fonction dérivé, 217
Bloc fonction élémentaire, 216
Bloc fonction élémentaire défini par l'utilisateur, 218
Boucles, 228
DFB, 217
EFB, 215
EN, 218
ENO, 218
Etablissement du programme, 233
FFB, 215
Flux de données, 228
flux de données, 226
Fonction, 215
Fonction élémentaire, 215
Fonction élémentaire définie par l'utilisateur, 218
fonctions en ligne, 231
Génération de codes, 229
Liaison, 219

FBD
Objet texte, 222
ordre d'exécution, 226
Paramètres réels, 220
Touches courtes, 874
UDEFB, 218
FDB, 211
Fenêtre, 835
Fermer une colonne
LL984, 473
FFB
Appel, 383
FBD, 215
Générer, FBD, 225
générer, LD, 258
Insérer, FBD, 225
insérer, LD, 258
Lancement, 372, 376, 448, 452
LD, 247
mettre en place, 258
Modifier, FBD, 225
modifier, LD, 259
Placer, 225
Remplacer, FBD, 225
remplacer, LD, 259
Fichier ENC, 22, 709
Fichier EXEC
Momentum, 200
UC 424 02, 162
UC X13 0X, 162
Fichier Include
définition avancée de type de données, 589
Fichier INI
chemin d'accès aux DFB globaux, 1153
chemin d'accès aux fichiers d'aide, 1153
Configuration de format d'adresse de registre, 1152
configuration de l'affichage des événements, 1165
configuration de l'édition en ligne, 1158
Configuration de messages d'avertissement, 1159
configuration du format d'adresse, 1159
Fichier INI
définition de nom de projet, 1152
emplacement des DFB globaux lors de la lecture, 1153
exclusion des DFB globaux/locaux de l'enregistrement en ligne, 1153
généralités, 1150, 1164
lecture de DFB globaux, 1153
paramètres d'impression, 1151
paramètres de l'éditeur de données, 1161
paramètres de l'enregistrement en ligne, 1166
paramètres de sécurité, 1160
paramètres du menu Tools, 1162
paramètres pour la section LD, 1157
représentation de données internes, 1156
spécifique du projet, 1147
fichier INI
configuration de dépôt de variables, 1152
Fichier journal, 709
Fichier journal chiffré, 22
Fichiers d'aide
définir un chemin d'accès, 1153
Fichiers EXEC, 1143
Fichiers INI, 1147
CONCEPT.INI, 1149
NomDeProjet.INI, 1163
fin de session automatique, 145
Flux de données, 262
FBD, 228
Fonction
FBD, 215
LD, 248
Fonction élémentaire
FBD, 215
LD, 248
Fonction élémentaire définie par l'utilisateur
FBD, 218
LD, 251
Fonctionnement EN LIGNE
Préréglages, 657

Fonctions disponibles en mode EN LIGNE et HORS LIGNE, 95
Fonctions en ligne, 20, 649
FBD, 231
IL, 391, 395
IL/ST, 392
Remarques générales, 651
SFC, 314
ST, 459
Fonctions en ligne dans le configurateur, 95
Fonctions en-ligne
LD, 265
fonctions en-ligne
SFC, 316
Fonctions hors ligne dans le configurateur, 95
Forcer
E/S, 612
Format d'adresse du registre
configuration du fichier INI, 1152
Fréquence d'échantillonnage
sections d'événements de temporisation, 1173

G

Gamme d'automate
Performances, 807
Généralités
connexion de l'API, 654
Extenseur d'arrière-plan, 128
Génération de code
IL, 388
ST, 457
Génération de codes
FBD, 229
LD, 263
Générer
FFB, FBD, 225
FFB, LD, 258

I

Icônes, 855, 857, 858, 859, 860, 862, 863, 864, 865, 866, 867
Icônes_Navigateur de projet, 867
IL, 325
Animation, 392, 395
Appel Bloc, 375
Barre d'outils, 863
Elaboration de programme, 397
Fonctions en ligne, 391, 392, 395
Génération de code, 388
Instruction, 329, 330
Modificateur, 334
Opérandes, 332
Opérateurs, 336, 345
Repère, 339
Touches courtes, 871
Vérification de la syntaxe, 386
IL-commande
Addition, 359
Charger, 346
Comapraison, 364, 367
Comparaison, 363, 365, 366, 368
Division, 362
Enregistrer, 347
ET logique, 352
Initialiser, 350
Lancement DBF, 372
Lancement de bloc fonction, 372
Multiplication, 361
Saut vers étiquette, 369
Soustraction, 360
Valider, 348
Import, 713
Configuration API, 754, 755
Section, 723, 729, 739, 740, 741
Variables, 743, 744, 747, 751
Variables structureées, 747
Importation
affectation multiple des adresses, 752
remarques générales, 715
Section, 724, 735
Importer configuration de Profibus DP, 971
Importer la configuration INTERBUS, 994
Impression des sections, 763
Imprimer
Paramètres dans le fichier INI, 1151
INC
fichier Include, 589
Informations ASCII, 116

Insérer
FFB, FBD, 225
FFB, LD, 258
Installation des pilotes Modbus Plus
Windows 98/2000/NT, 1054
Installation du SA85/PCI85
environnement Modbus Plus, 1048, 1052
Windows 98/2000/XP, 1048
Windows NT, 1052
Installation et commande des API, 669
Installer chargeables, 66
installer et actionner l'API
Remarques générales, 670
Installer et commander API, 669
Installer le fichier EXEC, 1143
Instruction
IL, 329, 330
Instructions
ST, 429, 430
Instructions chargeables, 106
Atrium, 207
Compact, 188
UC 424 02, 168
UC 434 12, 176
UC 534 14, 176
UC X13 0X, 168
Interaction avec la boîte de dialogue
LL984, 469

J

jonction alternative, 291
jonction simultanée, 293
Journalisation
fichier journal, 709
Journalisation chiffrée
fichier ENC, 709
Journalisation de l'accès en écriture à l'API, 709

L

Ladder Logic 984, 463
Lancement
DFB, 372, 376, 448
FFB, 372, 376, 448, 452
Langage des blocs fonctions, 211
Langages de programmation, 15
LD, 237
Animation, 265
Appeler macro, 556
Barre d'outils, 862
Bloc fonction, 249
Bloc fonction dérivé, 250
Bloc fonction élémentaire, 249
Bloc fonction élémentaire défini par l'utilisateur, 251
Bobine – Front négatif, 245
Bobine – Front positif, 245
Bobine – Initialiser, 246
Bobine – Inversée, 245
Bobine – Réinitialiser, 246
Bobines, 244
Boucles, 262
Contact de fermeture, 242
Contact de rupture, 242
Contacts, 242, 243
EFB, 247
EN, 252
ENO, 252
Etablissement du programme, 268
FFB, 247
flux de données, 260
Fonction, 248
Fonction élémentaire, 248
Fonction élémentaire définie par l'utilisateur, 251
Fonctions en-ligne, 265
Génération de codes, 263
liaison, 253
Objet texte, 256
ordre d'exécution, 260
Paramètres réels, 254
Touches courtes, 878
UDEFB, 251
Lecture de DFB globaux
Paramètres dans le fichier INI, 1153
Lecture de l'API, 698
Liaison
FBD, 219
LD, 253

Liaison réseau
Modbus, 658
Modbus Plus, 659
TCP/IP, 666
Liaison réseau Modbus, 658
Liaison réseau Modbus Plus, 659
Liaison réseau TCP/IP, 666
Lier
API, 653
Lire API, 698
Liste d'instructions, 325
Littéraux, 49
LL984, 463
Annuler, 472
Barre d'outils, 864
Décalage de références, 471
Edition, 472
Edition Réseaux, 474
Exigences, 468
Fermer une colonne, 473
Interaction avec la boîte de dialogue, 469
Mode combination, 489
Modes de programmation, 489
Modifier, 468
Navigation, 468
Ouvrir une colonne, 473
Ouvrir une ligne, 473
Première mise en service Momentum, 1083, 1099, 1120, 1139
Programmation directe, 489
Recherche en ligne, 477
Références, 470
Remplacer références, 477
Réseau d'équation, syntaxe et sémantique, 485
Réseau d'équations, 481, 482
Restriction en ligne, 469
Section, 465
Segment, 465
Sélectionner, 473
Sous-programmes, 479
Touches courtes, 885
Tracer, 477
Variables, 470
Zoom DX, 476
Zoom référence, 475

M

Macro, 535, 538
Aide sur contexte, 545
Appel à partir de FBD, 556
Appel à partir de LD, 556
Appel à partir de SFC, 553
Créer, 547
Documentation, 759
Effacer, 771
globale, 540
locale, 540
Marque d'échange, 542
Macro globale, 540
Macro locale, 540
Macros
Convertir, 1023
Marque d'échange
Macro, 542
Matériel
Performances, 807
Mémoire, 149
Optimisation de la mémoire ne dépendant pas de l'automate, 155
Optimiser, 154
Structure, 151
Mémoire API et optimisation
Atrium, 205
Compact, 185
Momentum, 197
Mémoire de l'API
Optimisation de la mémoire ne dépendant pas de l'automate, 155
Optimiser, 154
Structure, 151
Mémoire de l'API et optimisation
Quantum, 159, 173
Mémoire de l'API, 149
Mémoire et optimisation
Atrium, 205
Compact, 185
Momentum, 197
Quantum, 159, 173
Mémoire étendue, 166
Messages ASCII, 71

Mettre en place
FFB, LD, 258
Modbus
Première mise en service avec l'EXECLoader, 1067
Première mise en service avec le DOS-Loader, 1089
Première mise en service Compact, 1073, 1093
Première mise en service Momentum, 1078, 1083, 1096, 1099
Première mise en service Quantum, 1068, 1090
Modbus Plus
Pilote Remote MBX, 1057
Pilote Virtual MBX, 1055
Première mise en service Atrium, 1112, 1133
Première mise en service avec l'EXECLoader, 1103
Première mise en service avec le DOS-Loader, 1125
Première mise en service Compact, 1108, 1130
Première mise en service Momentum, 1116, 1120, 1136, 1139
Première mise en service Quantum, 1104, 1126
restriction d'accès en écriture, 145
ModConnect, 1027
Modes de programmation
LL984, 489
Modificateur
IL, 334
modification d'états de signal d'une variable localisée
éditeur de données de référence, 616
Modifications, chargement, 695
Modifié, 654
Modifier
bobine, LD, 259
contact, LD, 259
FFB, FBD, 225
FFB, LD, 259
LL984, 468
SFC, 295
Modifier la répartition E/S
Extenseur d'arrière-plan, 129
Modsoft
Compatibilité des fonctions, 1046
Conversion, 1037
Références, 1043
Momentum
Optimiser la mémoire, 197

N

Navigateur de projet, 573
barre d'outils, 867
Navigation
LL984, 468
Nom de projet.INI, 1147
NOM/NOE
blocage d'accès en écriture, 145
NomDeProjet.INI, 1163
configuration de l'affichage des événements, 1165
généralités, 1164
paramètres de l'enregistrement en ligne, 1166
Noms
Editeur de types de données, 597
Type de données dérivé, 597

O

Objet texte
FBD, 222
LD, 256
SFC, 294
Objets
insérer, LD, 258
SFC, 277
Opérandes
IL, 332
ST, 403
Opérateurs
IL, 336, 345
ST, 405, 409
Opération de lecture du MSTR, 146

Optimiser
Mémoire de l'API, 154
Optimisation de la mémoire ne dépendant pas de l'automate, 155
Options diverses, 124
Options pour l'impression des sections, 763
Ordonnanceur de segments, 109
Ordre d'exécution
sections d'événements de temporisation, 1179
ordre d'exécution
LD, 260
Ordre d'exécution
FBD, 226
section, 55
Outil d'affichage, 709
Outils (Tools), 24
Ouvrir
Projet, 846
projet, 846
Ouvrir une colonne
LL984, 473
Ouvrir une ligne
LL984, 473

P

Paramétrer interface ASCII, 122
Paramétrer interface Modbus, 122
Paramétrer interfaces
Interface ASCII, 122
Interface Modbus, 122
Paramétrer/modifier mot de passe API, 680
Paramètres d'export INTERBUS dans CMD, 989
Paramètres de liaison automatique, 846
paramètres de sécurité Quantum, 145
Paramètres réels
FBD, 220
LD, 254
Parcoureur de projet
Commande par clavier, 578
Commande par la souris, 578
Partition de mémoire API, 105
Partition mémoire, 66
Passage, 283
Performances
Gamme d'automate, 807
Matériel, 807
Phase
sections d'événements de temporisation, 1175
Pilote destiné à la connexion entre des adaptateurs d'interface hôte ModConnect et des applications 32 bits sous Windows 98/2000/NT
Pilote MBX, 1056
Pilote Ethernet MBX
Pilote pour la fonction Modbus Plus via TCP/IP, 1058
Pilote MBX
Pilote destiné à la connexion entre des adaptateurs d'interface hôte ModConnect et des applications 32 bits sous Windows 98/2000/NT, 1056
Pilote pour connexion distante (Remote)
Pilote Remote MBX, 1057
Pilote pour la fonction Modbus Plus via TCP/IP
Pilote Ethernet MBX, 1058
Pilote pour permettre l'exécution d'applications 16 bits sous Windows 98/2000/NT
Pilote Virtual MBX, 1055
Pilote Remote MBX
Modbus Plus, 1057
Pilote Virtual MBX
Modbus Plus, 1055
Placer
FFB, FBD, 225
Plan de contact, 237
Préférences Modbus Plus
Installation des pilotes Modbus Plus sous Windows 98/2000/NT, 1054
Première mise en service
Préréglages pour Modbus, 1061
Préréglages pour Modbus Plus, 1047
Première mise en service Atrium
DOS-Loader, 1133
EXECLoader, 1112
Modbus Plus, 1112, 1133
Première mise en service avec

l'EXECLoader
Modbus, 1067
Modbus Plus, 1103
Première mise en service avec le DOS-Loader
Modbus, 1089
Modbus Plus, 1125
Première mise en service Compact
DOS-Loader, 1093, 1130
EXECLoader, 1073, 1108
Modbus, 1073, 1093
Modbus Plus, 1108, 1130
Première mise en service Momentum
DOS-Loader, 1096, 1099, 1136, 1139
EXECLoader, 1078, 1083, 1116, 1120
Modbus, 1078, 1083, 1096, 1099
Modbus Plus, 1116, 1120, 1136, 1139
Première mise en service Quantum
DOS-Loader, 1090, 1126
EXECLoader, 1068, 1104
Modbus, 1068, 1090
Modbus Plus, 1104, 1126
Préréglages Modbus
Etablir la connexion matérielle, 1065
Problèmes de transmission, 1066
Réglage d'interface dans Windows NT, 1064
Préréglages Modbus Plus
Etablir la connexion matérielle, 1060
Préréglages pour fonctionnement EN LIGNE, 657
Préréglages pour Modbus
Première mise en service, 1061
Préréglages pour Modbus Plus
Première mise en service, 1047
Priorité
sections d'événements d'E/S, 1192
Problèmes de transmission
Préréglages Modbus, 1066
Procédez ainsi lors de la configuration, 91
PROFIBUS
Spécifier les unités couplées, 117
Programmation, 9
Programme
Créer, 61
Etat, 838
Programme
Structure, 41
structure, 42
traitement, 42
Programmes, 49
Programmes utilitaires, 24
Projet
Appeler, 846
appeler, 846
archiver, 768
Créer, 61
Documentation, 759
Effacer, 771
Ouvrir, 846
ouvrir, 846
Protéger, 802
Structure, 41
structure, 42
traitement, 42
Projets
Convertir, 1023
propriétés d'étape
traiter, 299
Protection d'accès aux données, 70
Protection des données dans la mémoire de signal, 121
Protection du mot de passe, 788, 790, 801
Protection par mot de passe, 787, 799
Protéger
DFB, 802
Projet, 802

Q

Quantum
Optimiser la mémoire, 159, 173
Quantum – Commande INTERBUS, 951

R

Raccourcis clavier, 869, 870, 871, 874, 878, 885
RDE, 611
Réactiver l'enregistrement Flash, 679
Recherche en ligne
LL984, 477

Recherche et insertion
noms de variables et d'adresses, 568
Recherche et insertion de noms de variables et d'adresses, 568
rechercher et remplacer
des noms de variables et des adresses, 564
Rédaction de programme
ST, 461
Références
LL984, 470
Réglage d'interface dans Windows NT
Préréglages Modbus, 1064
Règlage en ligne
Remarques générales, 670
Réglages d'exportation de Profibus DP dans SyCon, 965
Relier avec le simulateur CEI (32 bit), 667
Relier le simulateur CEI (32 bit), 667
Remarques générales, 1
Chargement d'un projet, 690
Extraire les informations de processus, 684
Fonctions en ligne, 651
Remarques générales sur l'éditeur de données de référence, 612
Remarques générales sur l'éditeur de variables, 560
Remarques générales sur la configuration API, 90
Remarques générales sur la configuration en mode HORS LIGNE et EN LIGNE, 94
Remarques générales sur la configuration matérielle, 89
Remarques générales sur le règlage en ligne, 670
Remarques générales sur les fonctions en ligne, 651
Remplacement de variables
Editeur de données de référence, 625
Remplacer
bobine, LD, 259
contact, LD, 259
FFB, FBD, 225
FFB, LD, 259
Remplacer références
LL984, 477
Repère
IL, 339
Réseau d'équation, syntaxe et sémantique
LL984, 485
Réseau d'équations
LL984, 481, 482
Restriction en ligne
LL984, 469
Restrictions
LL984, 466

S

saut
SFC, 288
saut de chaîne, 288
Sauts de page pour les sections, 763
Scrutateur d'E/S / Ethernet
comment utiliser le scrutateur d'E/S / Ethernet, 142
Scrutateur E/S / Ethernet
Configurateur, 138
Section, 54
Animation, 703
désactiver, 56
Etat, 838
Export, 717
Import, 723, 729, 739, 740, 741
Importation, 724, 735
LL984, 465
ordre d'exécution, 55
Section CEI
Animation, 704
Section d'événement d'E/S
manipulation, 1167
Section d'événement de temporisation
Manipulation, 1167
Section de transition, 285
Section LL984
Animation, 706
Sections d'evénements d'E/S
erreur d'exécution, 1193
Sections d'événements d'E/S, 1190
priorité, 1192

Sections d'événements de temporisation, 1172
  configurer la fréquence d'échantillonnage, 1173
  configurer la phase, 1175
  Exemples de paramètres définis, 1183
  ordre d'exécution, 1179
  système d'exécution, 1180
Sections d'interruption
  désactiver, 56
  EFBs, 1195
  erreur d'exécution, 1193
  Exemples de paramètres définis, 1183
  fréquence d'échantillonnage de sections d'événements de temporisation, 1173
  ordre d'exécution, 1179
  priorité, 1192
  sections d'événements d'E/S, 1190
  sections d'événements de temporisation, 1172, 1175
  système d'exécution, 1180
Sécurité, 788, 799
Sécurité Concept, 788
Security, 787, 790, 801, 802
Segment
  LL984, 465
Sélection automate, 99
Sélection d'UC pour le type d'automate, 100
Sélectionner
  LL984, 473
Séparateurs
  éditeur de types de données, 598
  type de données dérivé, 598
SFC
  'Variable SFCSTEP_STATE', 280
  'Variable SFCSTEP_TIMES', 279
  Action, 281
  actions, 301
  Animation, 314
  animation, 316
  Appeler macro, 553
  Apprendre temps de surveillance, 320
  Barre d'outils, 860
SFC
  caractères de définition, 304
  Chaîne, 316
  connextion, 287
  dérivation simultanée, 292
  Diagnostic de transition, 323
  divergence en OU, 289
  En ligne, 313
  Etape, 278
  Etape d'attente, 278
  Etape initiale, 278
  Fonctions en ligne, 314
  fonctions en- ligne, 316
  jonction alternative, 291
  jonction simultanée, 293
  Modifier, 295
  Objet texte, 294
  Objets, 277
  Passage, 283
  propriétés d'étapes, 299
  saut, 288
  Section de transition, 285
  Temporisation de l'étape, 278
  Temps de contrôle maximum, 278
  Temps de contrôle minimum, 279
  Temps de retard, 278
  Touches courtes, 874
  traiter, 296
  Transition, 283, 307
  Variable d'action, 281
  variable de transition, 286
Simulation, 773, 775, 777
Simuler
  API, 775, 777
Sorties ANY, 448
Sous-programmes
  LL984, 479
ST, 399
  Animation, 392
  Appel composant, 447
  Barre d'outils, 863
  Expressions, 402
  Fonctions en ligne, 392, 459

ST, 399
  Génération de code, 457
  Instructions, 429, 430
  Opérandes, 403
  Opérateurs, 405, 409
  Rédaction de programme, 461
  Touches courtes, 871
  Vérification de la syntaxe, 456
Station décentralisée d'E/S, 68
Statistiques mémoire, 686
Structure
  Mémoire de l'API, 151
  Programme, 41
  programme, 42
  Projet, 41
  projet, 42
Symbole, 855
Symbole de projet
  Créer, 846
  créer, 846
Symboles, 857, 858, 859, 860, 862, 863, 864, 865, 866, 867
Syntaxe
  Editeur de types de données, 591
  Type de données dérivé, 591
Système d'exécution
  sections d'événements de temporisation, 1180
Système de bus Ethernet
  Etablir connexion en ligne, 1021
Système de bus Ethernet (Momentum), 1008

T

TCP/IP
  Configuration réseau, 1009
Temporisation de l'étape, 278
Temps de contrôle maximum, 278
Temps de contrôle minimum, 279
Temps de retard, 278
Texte structuré, 399
Tools
  Paramètres dans le fichier INI, 1162
Touches, 855, 869, 870, 871, 874, 878, 885
Touches courtes, 869, 870, 871, 874, 878, 885
Touches rapides, 855
Tracer
  LL984, 477
Traitement
  programme, 42
  projet, 42
Traitement d'interruption, 1167
  généralités, 1169
Traitement des erreurs
  Extenseur d'arrière-plan, 130
Traitement LL984
  à vitesse optimisée, 675
Traitement LL984 à vitesse optimisée, 675
traiter
  actions, 301
  propriétés d'étapes, 299
  SFC, 296
Transition, 283
  Déclarer, 307
  Désignations pseudonymes, 309
  Editer, 307
Transmission globale de données
  Diffusion des E/S, 979
Transmission spécifique de données
  Diffusion des E/S, 981
Type de donné dérivé
  Commentaire, 600
  Eléments, 592
  Noms, 597
  Syntaxe, 591
Type de données dérivé, 581, 583, 584
  Clés, 593
  Global, 587
  Local, 587
  Occupation de la mémoire, 601
  séparateurs, 598
  utilisation, 603
Type de données dérivé local, 587
Types de données dérivés
  Export, 721
Types de fenêtres, 836

U

UDEFB
FBD, 218
LD, 251
Unité de fonctions dérivée, 494

V

Variable
Export, 721
Variable d'action, 281
variable de transition, 286
Variables, 49
comportement de démarrage, 51
Editeur de messages ASCII, 632
Import, 743, 744, 747, 751
LL984, 470
Variables globales dans les DFB, 507
variables localisées
modification d'états de signal dans RDE, 616
Variables structureées
Import, 747
Variables VARINOUT, 500
Vérification de la syntaxe
IL, 386
ST, 456
Verrouillage inconditionnel d'une section, 621

W

Windows, 833
Barre d'état, 838
Boîtes de dialogue, 843
Bouton d'option, 844
Boutons de commande, 844
Case à cocher, 845
Commandes, 841
Eléments de fenêtre, 838
Fenêtre, 835
Types de fenêtres, 836
Zones de liste, 844
Zones de texte, 844

Z

Zoom DX
LL984, 476
Zoom référence
LL984, 475